



A probabilistic graphical model approach to stochastic multiscale partial differential equations



Jiang Wan^a, Nicholas Zabaras^{a,b,*}

^a Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, Cornell University, 101 Frank H.T. Rhodes Hall, Ithaca, NY 14853-3801, USA

^b Center for Applied Mathematics, Cornell University, 657 Frank H.T. Rhodes Hall, Ithaca, NY 14853, USA

ARTICLE INFO

Article history:

Received 4 December 2012

Received in revised form 30 March 2013

Accepted 7 May 2013

Available online 24 May 2013

Keywords:

Bayesian

Probabilistic graphical models

Uncertainty quantification

Multiscale modeling

Sequential Monte Carlo

Stochastic partial differential equations

Belief propagation

ABSTRACT

We develop a probabilistic graphical model based methodology to efficiently perform uncertainty quantification in the presence of both stochastic input and multiple scales. Both the stochastic input and model responses are treated as random variables in this framework. Their relationships are modeled by graphical models which give explicit factorization of a high-dimensional joint probability distribution. The hyperparameters in the probabilistic model are learned using sequential Monte Carlo (SMC) method, which is superior to standard Markov chain Monte Carlo (MCMC) methods for multi-modal distributions. Finally, we make predictions from the probabilistic graphical model using the belief propagation algorithm. Numerical examples are presented to show the accuracy and efficiency of the predictive capability of the developed graphical model.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Physical systems generally have inherent randomness which could result from uncertainties in boundary or initial conditions, material heterogeneities and so on. It is therefore necessary to include parameters characterizing these uncertainties into the model system. In order to study the propagation of uncertainties from the parameter space to the response space, stochastic differential equations are constructed such that both input parameters and model responses are modeled as random variables. The scope of this work focuses on stochastic partial differential equations (SPDEs).

Over the past few decades, many methods and algorithms have been developed to solve SPDEs. The most celebrated one is the Monte Carlo (MC) method. As a sampling method, the deterministic solver is called for each realization of the stochastic input for one to obtain the statistics of the solution. The convergence rate does not depend on the dimension of the parameter space, but is of order $\mathcal{O}(n^{-1/2})$ with n realizations. To accelerate convergence, quasi Monte Carlo methods and several efficient sampling techniques have been developed as alternatives. Another group of methods refers to nonsampling approaches, typically perturbation algorithms based on a series representation of the stochastic solution. These methods are limited to small fluctuations and low-order statistics of the solution. Recently, many research efforts have been devoted to the study of schemes based on spectral representation of the stochastic solution [1]. For example, the well-established stochastic Galerkin method approximates the solution in a multivariate polynomial space or in anisotropic tensor product polynomial spaces [2]. Stochastic collocation methods using sparse grids based on the Smolyak algorithm

* Corresponding author at: Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, Cornell University, 101 Frank H.T. Rhodes Hall, Ithaca, NY 14853-3801, USA. Tel.: +1 607 255 9104; fax: +1 607 255 9410.

E-mail addresses: nzabaras@gmail.com, zabaras@cornell.edu (N. Zabaras).

URL: <http://mpdc.mae.cornell.edu/> (N. Zabaras).

[3] have a weaker dependence on the dimensionality of the problem and recently have been applied extensively to various uncertainty quantification problems [4–8].

In spite of the tremendous progress in solving SPDEs, the curse of dimensionality remains a long-standing challenge. For conventional stochastic Galerkin methods, the computational cost depends on the number of expansion terms which grows exponentially as a function of the dimensionality of stochastic input space [9]. Stochastic collocation methods can achieve fast convergence rate by taking advantage of multidimensional polynomial interpolation. However, the number of collocation points required to achieve sufficient accuracy increases exponentially for high-dimensional problems [10]. Therefore, many efforts have been devoted to stochastic methods that deal with high-dimensions. In [7], an adaptive sparse grid collocation (ASGC) method is proposed such that the collocation points are selected automatically based on the smoothness of the stochastic domain as detected by the magnitude of the hierarchical surpluses. The ASGC can successfully solve stochastic elliptic problems up to 100 dimensions when not all stochastic dimensions are equally important [8]. However, the convergence rate deteriorates even for problems with moderate input dimensionality when all stochastic dimensions are equally weighted. Another direction is to decompose the original problem into sub-problems with low-dimensional input. A typical example is the high-dimensional model representation (HDMR) techniques that capture the high-dimensional relationships between input and output model variables and generate a collection of low-dimensional sub-problems in stochastic space [11]. Recent progress in HDMR can be found in [8,12]. In [9], a low-rank separated representation of the solution to SPDEs with high-dimensional inputs is obtained using an alternating least-squares approach.

More challenges arise when multiscale phenomena are taken into account in high-dimensional stochastic problems. In such cases, information across scales contains a certain level of uncertainties but assessment of uncertainty propagation often leads to large computational cost. Let us take, for example, fluid flow through porous media occurring from large geological scales down to microscopic scales. Full-scale spatial and temporal resolution simulations may require significant computational resources. Since the sample-based stochastic methods mentioned above, such as MC, ASGC, and HDMR, call the deterministic solvers repetitively, efficient solvers for multiscale partial differential equations are of great importance in reducing overall computational cost. For this purpose, computational techniques, such as the multiscale finite element (MsFEM) method [13–15], variational multiscale (VMS) method [16,17], the heterogeneous multiscale (HMM) method as well as their variants [10,18,19] and multigrid methods [20,21], have been developed to solve a coarse-scale problem that captures fine-scale effects without resolving all the fine-scale features.

So far, many efforts for solving multiscale stochastic problems focus on decoupling multiscale deterministic solvers from stochastic approaches. In this work, we propose a new scheme to quantify the uncertainties propagated in multiscale systems based on a probabilistic model of SPDE solutions. Inference problems can be solved directly on this probabilistic model without sampling-based methods or calling expensive deterministic solvers. The stochastic input and model responses are all treated as random variables. However, conventional regression models are inefficient or even impractical to represent their relationships when the stochastic input is in a high-dimensional space. This curse of dimensionality can be overcome by utilizing probabilistic graphical models which have been intensively studied and widely used in machine learning and Bayesian statistics for multivariate statistical modeling [22,23]. A probabilistic graphical model consists of a collection of nodes and edges that connect pairs of nodes. Each node represents a random variable or a group of random variables and the edges express probabilistic relationships between variables, e.g. conditional independence properties. By combining both graph theory and probability theory, the complicated relationships between all variables can be modeled explicitly and the resulting graph expresses a decomposition of a joint distribution as a product of functions of subsets of variables. Given the graph-based probabilistic model of model responses, efficient algorithms for inference on graphical models can be directly applied. If we treat the stochastic input as observed variables, the probabilistic model becomes a conditional distribution of model responses on input variables, which leads to a surrogate model. The predictions of model responses can be evaluated by inference algorithms on the graphical model associated with this surrogate model. If the stochastic input field also has an explicit graphical representation, we can directly estimate the marginal distributions of unobserved variables in the graphical model by integrating out all the other variables with an efficient algorithm.

This paper is organized as follows. In Section 2, we introduce the classical flow in random porous media benchmark multiscale problem to set forward the presentation of the methodology. In Section 3, we construct the framework of probabilistic modeling of coarse-scale model responses. Then we introduce a sequential Monte Carlo method to learn the hyperparameters in the graphical model from training data in Section 4. The belief propagation algorithm presented in Section 5 is utilized to predict model responses based on the probabilistic graphical model. Case studies on the stochastic porous media flow problem are provided in Section 6.

2. Problem definition

To model uncertainties in a physical system, we define a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ where Ω is a sample space, \mathcal{F} a σ -algebra of subsets of Ω and $\mathcal{P} : \mathcal{F} \rightarrow [0, 1]$ a probability measure. Let $D \subset \mathbb{R}^d$ be a fixed d -dimensional bounded domain with boundary ∂D . A general stochastic partial differential equation (SPDE) is formulated as

$$\mathcal{L}(\mathbf{x}, \omega; y) = 0, \quad \forall \mathbf{x} \in D, \quad (1)$$

with boundary conditions

$$\mathcal{B}(\mathbf{x}, \omega; y) = 0, \quad \forall \mathbf{x} \in \partial D, \quad (2)$$

where \mathcal{L} is a general differential operator, \mathcal{B} is a boundary operator, $\omega \in \Omega$ is an elementary event in the sample space and y is the model response.

We are interested in assessing macroscopic quantities from fine-scale information based on SPDEs with multiscale features. For demonstration of the approach, we consider single phase incompressible fluid flow in porous media where the length scale of permeability variation is orders of magnitude smaller than the characteristic length scale of the domain. The pressure h and velocity \mathbf{u} are characterized by the following equations [24]

$$\nabla \cdot \mathbf{u} = f, \quad \mathbf{u} = -K(\mathbf{x}, \omega) \nabla h, \quad \forall \mathbf{x} \in D, \quad (3)$$

with boundary conditions

$$h = \bar{h} \text{ on } \partial D_h, \quad \mathbf{u} \cdot \mathbf{n} = \bar{\mathbf{u}} \text{ on } \partial D_u, \quad (4)$$

with the assumptions that the effects of gravity, capillary pressure, and compressibility can be neglected and that the porosity is constant. For simplicity, we let the random permeability tensor $K(\mathbf{x}, \omega)$ be isotropic and use the log-permeability $a(\mathbf{x}, \omega) = \log(K)$ as it is commonly employed in geostatistical models.

Due to the multiscale nature of the problem, two sets of grids are defined. First, let us discretize the domain D into non-overlapping elements e_i and obtain a fine-scale grid $\mathcal{T}_h = \bigcup_{i=1}^{N_h} e_i$, where N_h is the number of fine elements. In this work, the global log-permeability is defined as $\mathbf{a} \equiv \{a_i(\omega)\}_{i=1}^{N_h}$ where $a_i(\omega)$ is the log-permeability on the i th fine-scale element. Then we define the skeleton of the fine-scale partition, $\mathcal{SP}_h = \bigcup_{a=1}^{M_h} v_a$, where v_a denote element faces and M_h is the total number of faces. In a multiscale problem, a coarse-scale partition of the same domain is proposed. Denote this partition as $\mathcal{T}_c = \bigcup_{i=1}^{N_c} E_i$ where E_i are coarse elements. We also denote the associated skeleton of the coarse-scale discretization by $\mathcal{SP}_c = \bigcup_{a=1}^{M_c} \Lambda_a$. Here, N_c is the number of coarse elements and M_c is the number of coarse element faces denoted by Λ_a . The two partitions, \mathcal{T}_h and \mathcal{T}_c , are nested. Fig. 1 shows a fine grid (finer lines) and a corresponding coarse grid (heavier lines). The log-permeability on the k th coarse element is a random vector denoted by

$$\mathbf{a}_k \equiv \{a_j(\omega)|e_j \subset E_k\}, \quad (5)$$

which is referred to as local features and $\mathbf{a}_k \subset \mathbf{a}$.

3. Probabilistic model of responses

A main objective of multiscale modeling is assessing coarse-scale model responses. However, conventional methods such as collocation methods suffer from the curse of dimensionality when the stochastic input \mathbf{a} is in a high-dimensional space. In order to overcome this difficulty, a probabilistic graphical model representation of multiscale SPDE is directly constructed in this paper and probability theory is applied to this model to make predictions of model responses.

Without loss of generality, let us suppose that we are interested in model responses on a set of regularly distributed points in the coarse-grid. The coarse-scale model responses associated with these points are denoted by a vector $\mathbf{Y} = (y_1, \dots, y_n)^T$ where y_i stands for model responses. In uncertainty quantification, we aim at estimating the probability distribution of \mathbf{Y} , i.e. $p(\mathbf{Y})$. In a probabilistic framework, this multivariate joint distribution is given by

$$p(\mathbf{Y}) = \int p(\mathbf{Y}, \mathbf{a}) d\mathbf{a} = \int p(\mathbf{Y}|\mathbf{a})p(\mathbf{a})d\mathbf{a}, \quad (6)$$

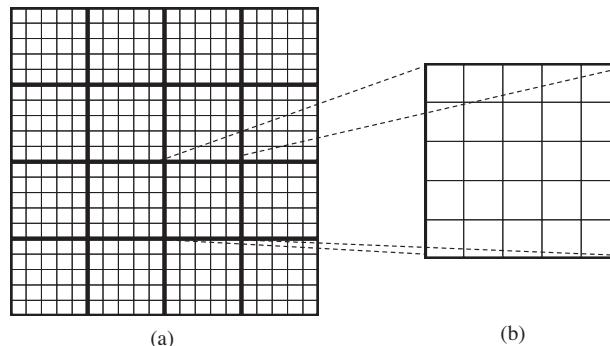


Fig. 1. Schematic of the domain partition: (a) fine- and coarse-scale grids and (b) fine-scale local region in one coarse element.

which is based on the knowledge of the stochastic input model and the probabilistic dependence between output and input. In theory, the mapping from \mathbf{a} to \mathbf{Y} is deterministic given a forward model, which implies that $p(\mathbf{Y}|\mathbf{a})$ degenerates to delta functions. However, from the perspective of Bayesian statistics, we learn the relationship between input and output completely from training data $\mathcal{D} = \{\mathbf{a}^{(t)}, \mathbf{Y}^{(t)}\}$. As a result, many other sources of uncertainties, such as the lack of knowledge on forward models and various modeling errors, come into the picture. Then Eq. (6) can be approximated as

$$p(\mathbf{Y}|\mathcal{D}) = \int p(\mathbf{Y}|\mathbf{a}, \mathcal{D})p(\mathbf{a})d\mathbf{a}. \quad (7)$$

Actually, the training data affect the estimation of the relationship between \mathbf{a} and \mathbf{Y} through unknown model parameters, Θ^* , i.e. $p(\mathbf{Y}|\mathbf{a}, \mathcal{D}) \equiv p(\mathbf{Y}|\mathbf{a}, \Theta^*(\mathcal{D}))$. For notational convenience, we will ignore Θ^* temporarily. But it will be brought into the framework later in Section 3.2 and will be learned from training data in Section 4.

In many practical applications, a stochastic input model can be learned from observation data and thus $p(\mathbf{a})$ is assumed to be known. However, \mathbf{a} is generally in a high-dimensional space. Although various model reduction techniques might map \mathbf{a} to a lower-dimensional space, it is still challenging to construct an accurate probabilistic model of $p(\mathbf{Y}|\mathbf{a})$ using conventional methods such as kernel density estimation. Therefore, we consider a conditional random field (CRF) representing $p(\mathbf{Y}|\mathbf{a})$ with a Gibbs distribution as:

$$p(\mathbf{Y}|\mathbf{a}) \propto \exp(-\mathcal{E}(\mathbf{Y}; \mathbf{a})), \quad (8)$$

where $\mathcal{E}(\mathbf{Y}; \mathbf{a})$ is an ‘energy’ function. Let I denote the index set of elements in \mathbf{Y} . Then, the energy function can be written in the following general form

$$\mathcal{E}(\mathbf{Y}; \mathbf{a}) \approx \sum_{i \in I} \phi_i^{(1)}(y_i, \mathbf{a}) + \sum_{(i,j) \in I \times I} \phi_{ij}^{(2)}(y_i, y_j, \mathbf{a}) + \dots + \phi_I^{(|I|)}(\mathbf{Y}, \mathbf{a}),$$

where $\phi^{(n)}$ are feature functions of n variables in \mathbf{Y} . In this work, we approximate the energy function by ignoring high-order interactions among the model responses. Hence, only $\phi^{(1)}$ and $\phi^{(2)}$ are retained:

$$\mathcal{E}(\mathbf{Y}; \mathbf{a}) \approx \sum_{i \in I} \phi_i(y_i, \mathbf{a}) + \sum_{(i,j) \in I \times I} \phi_{ij}(y_i, y_j, \mathbf{a}). \quad (9)$$

For notational convenience, the superscripts in potential functions are omitted. Since only up to pairwise interactions between variables in \mathbf{Y} are considered in this framework, we further assume that the conditional random field $p(\mathbf{Y}|\mathbf{a})$ is a Gaussian Markov random field which can be formulated by [22,25]

$$p(\mathbf{Y}|\mathbf{a}) \propto \exp \left(-\sum_i f_i(\mathbf{a})y_i - \sum_i \sum_j f_{ij}(\mathbf{a})y_i y_j \right), \quad (10)$$

where $f_i(\cdot)$ and $f_{ij}(\cdot)$ are functions of the stochastic input.

So far, a global approximation of $p(\mathbf{Y}|\mathbf{a})$ has been proposed. However, this is of little practical use. Due to the high-dimensional stochastic input, it is difficult to estimate the functions $f_i(\mathbf{a})$ and $f_{ij}(\mathbf{a})$. Moreover, it is impractical to make predictions directly from this probabilistic model of \mathbf{Y} . Therefore, the global problem is decomposed into lower-dimensional local sub-problems. Then the approximation of conditional distribution proposed in Eq. (10) is applied on each sub-problem. A graphical model associated with the probabilistic model is also presented for making predictions of model responses.

3.1. Brief introduction to probabilistic graphical models

A graph $G = (V, E)$ comprises a collection of nodes V connected by links E . It combines both probability theory and graph theory for multivariate statistical modeling [22]. In a graphical model, each node represents a random variable or a group of random variables, and the links express dependence between these variables or groups. Commonly used graphical models include Bayesian networks, undirected graphical models and factor graphs. They define the dependence relationships between random variables in different ways.

The main advantage of a graphical model is its ability to factorize the joint distribution over all of the random variables into a product of factors each depending only on a subset of the variables according to the structure of the underlying graph. In an undirected graphical model, which has undirected links, a clique is defined as a subset of nodes such that there exists a link between any pair of nodes in the subset. A maximal clique is a clique that would cease to be a clique when any other nodes are included. Suppose the graphical model is for random variables \mathbf{x} , i.e. $V = \{x_1, \dots, x_n\}$. Then, the joint distribution is factorized as a product of potential functions μ_c over maximal cliques

$$p(\mathbf{x}) \propto \prod_c \mu_c(\mathbf{x}_c),$$

where C denotes a maximal clique and \mathbf{x}_c the variables in that clique.

An undirected graphical model can be converted to a factor graph which provides an efficient way of describing the factorization properties of large graphs [22]. The factor graph, denoted by $G = (V, E, F)$, decomposes a joint probability explicitly

by introducing additional factor nodes (functions) F . The joint probability is assumed to be proportional to the product of all factor nodes in the graph. For example, consider the factorization

$$p(\mathbf{x}) \propto \prod_{j=1}^m \mu_j(\mathbf{x}_j),$$

where $\mathbf{x}_j \subset \mathbf{x}$. In the factor graph for this joint distribution, the factor nodes $F = \{\mu_1, \dots, \mu_m\}$. A factor node, $\mu_i(\mathbf{x}_i)$, is linked to every element in \mathbf{x}_i . In a typical factor graph, the vertices indicate random variables (or vectors) and squares indicate the factor nodes (see Fig. 3(b)). The factor nodes contain more detailed information about underlying factorization of cliques in an equivalent undirected graphical model. Based on prior knowledge of interacting variables, the factor nodes are properly defined to enable efficient computation of marginal distributions over the joint distribution. This feature will be discussed in the inference problem in Section 5. For more details on undirected graphical models, factor graphs and inference on graphical models, the reader can refer to [22,23,26].

3.2. Probabilistic graphical model for multiscale SPDEs

In this section, a probabilistic model for $p(\mathbf{Y}|\mathbf{a})$ is constructed with the help of a graphical model. In a multiscale system with stochastic input, the corresponding graph includes nodes for two types of random variables, stochastic input \mathbf{a} and output (model responses) \mathbf{Y} . As discussed in Section 2, the stochastic input \mathbf{a} is equivalent to $\{\mathbf{a}_1, \dots, \mathbf{a}_{N_c}\}$ in which \mathbf{a}_k denotes the local fine-scale features on coarse element E_k and N_c the number of coarse elements. In fluid flow through porous media, the pressure plays an important role in Darcy's law. As in the mixed finite element method, we assume a constant pressure, h_k , on each coarse element. The flow in a coarse element interacts with the flow at the neighboring elements through the adjoint edges. Thus the flux on middle points of edges of coarse elements is also of interest. Finally, the target model responses include the pressure on each coarse element and the fluxes on the middle points of the edges of the coarse elements, i.e. $\mathbf{Y} = \{\mathbf{u}, \mathbf{h}\}$ where $\mathbf{h} = \{h_1, \dots, h_{N_c}\}$ denote the pressure variables and $\mathbf{u} = \{u_1, \dots, u_{M_c}\}$ the fluxes. The spatial distribution of nodes for output and local input features in a graphical model is depicted in Fig. 2.

As there is little prior information on the relationship between the random variables $\{\mathbf{u}, \mathbf{h}\}$ and \mathbf{a} , all nodes in each coarse element are mutually linked (as shown in Fig. 2(b)). Note that the random vector \mathbf{a}_k is treated as a single node in the graph. In order to simplify the graph structure, the spatial correlations between the model responses are taken into account. Each response is only correlated to the response of its neighboring nodes in the same coarse element and long distance interactions are ignored. In this way, the variables on a coarse element, $\{u_{i \in I_k}, h_k, \mathbf{a}_k\}$, form a maximal clique in an undirected graphical model. Here, I_k is the index set of responses in element E_k . By the Hammersley–Clifford theorem [27,28],

$$p(\mathbf{u}, \mathbf{h}|\mathbf{a}) \propto \prod_k q_k(u_{I_k}, h_k; \mathbf{a}_k), \quad (11)$$

where $q_k(\cdot)$ is the potential function of the maximal clique on E_k . Thus the global joint distribution is factorized into local potential functions on coarse elements. From the conditional random field defined in Eq. (8), $p(\mathbf{u}, \mathbf{h}|\mathbf{a})$ can be reformulated as

$$p(\mathbf{u}, \mathbf{h}|\mathbf{a}) \propto \exp \left(-\sum_k \mathcal{E}_k(u_{I_k}, h_k; \mathbf{a}_k) \right), \quad (12)$$

such that $q_k(u_{I_k}, h_k; \mathbf{a}_k) := \exp(-\mathcal{E}_k(u_{I_k}, h_k; \mathbf{a}_k))$. Then Eq. (9) is applied to approximate these local energy functions

$$\mathcal{E}_k(u_{I_k}, h_k; \mathbf{a}_k) \approx \sum_{i \in I_k} \phi_{k,i}(u_i, \mathbf{a}_k) + \sum_{(i,j) \in I_k \times I_k, i \neq j} \phi_{k,ij}(u_i, u_j, \mathbf{a}_k) + \phi_{k,0}(h_k, \mathbf{a}_k) + \sum_{i \in I_k} \phi_{k,i0}(u_i, h_k, \mathbf{a}_k), \quad (13)$$

where from the Gaussian Markov field approximation in Eq. (10):

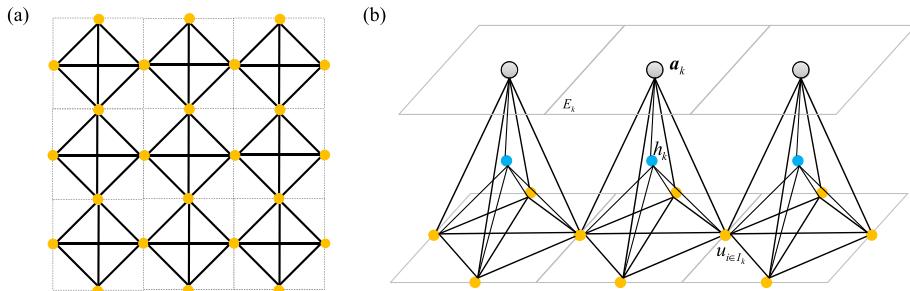


Fig. 2. (a) Graphical representation of the relationships between model responses, (b) undirected graph for the stochastic input \mathbf{a} and model responses \mathbf{Y} .

$$\begin{aligned}
\phi_{k,i}(u_i, \mathbf{a}_k) &:= f_{k,i}(\mathbf{a}_k)u_i + f_{k,ii}(\mathbf{a}_k)u_i^2, \\
\phi_{k,ij}(u_i, u_j, \mathbf{a}_k) &:= f_{k,ij}(\mathbf{a}_k)u_iu_j, \quad i \neq j, \\
\phi_{k,0}(h_k, \mathbf{a}_k) &:= f_{k,0}(\mathbf{a}_k)h_k + f_{k,00}(\mathbf{a}_k)h_k^2, \\
\phi_{k,i0}(u_i, h_k, \mathbf{a}_k) &:= f_{k,i0}(\mathbf{a}_k)u_ih_k.
\end{aligned} \tag{14}$$

Note that the potential $\phi_{k,ij}$ with $i = j$ is accounted into $\phi_{k,i}$. The functions, $f(\cdot)$, measure the influence of local features on model responses and for stationary permeability are assumed identical on different coarse elements. Nonparametric models are adopted for all functions such that

$$f_{k,\cdot}(\mathbf{a}_k) \equiv f_{k,\cdot}(\mathbf{a}_k; \theta_k) = \theta_{k,\cdot}^{(1)} + \sum_{t=2}^r \theta_{k,\cdot}^{(t)} \zeta_t(\mathbf{a}_k), \tag{15}$$

where we choose unnormalized Gaussian kernels $\zeta_t(\mathbf{a}_k) = \exp\left(-\frac{\|\mathbf{a}_k - \bar{\mathbf{a}}_t\|^2}{\sigma_\zeta^2}\right)$ (chapter 6 in [29]). The hyperparameters $\theta_k = \{\theta_{k,\cdot}^{(t)}\}$ are fixed and will be learnt from training data. We also use $\Theta = \bigcup_k \theta_k$ to denote all hyperparameters in the probabilistic model.

Given a set of samples of input $\{\mathbf{a}_k^{(n)}\}_{n=1}^N$ and specifying the number of kernels r , the centers of Gaussian kernels, $\{\bar{\mathbf{a}}_t\}_{t=1}^r$, are determined using K-means. A typical choice of the kernel width σ_ζ is the average minimum distance between two realizations in the input space, i.e.

$$\sigma_\zeta^2 = \frac{1}{N} \sum_{i=1}^N \min_{i \neq j} \|\mathbf{a}_k^{(i)} - \mathbf{a}_k^{(j)}\|^2. \tag{16}$$

Combining Eqs. (12) and (13), the conditional distribution of model responses is formulated as

$$\begin{aligned}
p(\mathbf{u}, \mathbf{h} | \mathbf{a}, \Theta) \propto \prod_k \left(\prod_{i \in I_k} \exp(-f_{k,i}(\mathbf{a}_k; \theta_k)u_i - f_{k,ii}(\mathbf{a}_k; \theta_k)u_i^2) \cdot \prod_{(i,j) \in I_k \times I_k, i \neq j} \exp(-f_{k,ij}(\mathbf{a}_k; \theta_k)u_iu_j) \cdot \prod_{i \in I_k} \exp(-f_{k,i0}(\mathbf{a}_k; \theta_k)u_ih_k) \right. \\
\left. \cdot \exp(-f_{k,0}(\mathbf{a}_k; \theta_k)h_k - f_{k,00}(\mathbf{a}_k; \theta_k)h_k^2) \right).
\end{aligned} \tag{17}$$

Any realization of the stochastic input \mathbf{a} influences the energy function (Eqs. (12) and (13)) through the function values $f_{k,\cdot}(\mathbf{a}_k; \theta_k)$. The function $f_{k,\cdot}$ is a mapping $f_{k,\cdot} : \mathbf{a}_k \rightarrow \xi_{k,\cdot}$ from local features \mathbf{a}_k to a scalar variable $\xi_{k,\cdot}$. In other words, $f_{k,\cdot}$ projects the high-dimensional input into a low-dimensional space. Since these variables $\xi_{k,\cdot}$ are not directly observable, we call them hidden variables in the probabilistic model. The relationships between hidden variables and local features are as follows: $\xi_{k,i} = f_{k,i}(\mathbf{a}_k; \theta_k)$, $\xi_{k,ii} = f_{k,ii}(\mathbf{a}_k; \theta_k)$, $\xi_{k,ij} = f_{k,ij}(\mathbf{a}_k; \theta_k)$, $\xi_{k,0} = f_{k,0}(\mathbf{a}_k; \theta_k)$, $\xi_{k,00} = f_{k,00}(\mathbf{a}_k; \theta_k)$ and $\xi_{k,i0} = f_{k,i0}(\mathbf{a}_k; \theta_k)$, where the hyperparameters $\theta_k = \{\theta_{k,\cdot}^{(t)}\}$ were introduced in Eq. (15). The conditional distribution of $\{\mathbf{u}, \mathbf{h}\}$ can now be formulated in terms of the hidden variables ξ :

$$p(\mathbf{u}, \mathbf{h} | \xi) \propto \prod_k \left(\prod_{i \in I_k} \exp(-\xi_{k,i}u_i - \xi_{k,ii}u_i^2) \cdot \prod_{(i,j) \in I_k \times I_k, i \neq j} \exp(-\xi_{k,ij}u_iu_j) \prod_{i \in I_k} \exp(-\xi_{k,i0}u_ih_k) \cdot \exp(-\xi_{k,0}h_k - \xi_{k,00}h_k^2) \right). \tag{18}$$

According to the definition of the hidden variables, each of them is completely fixed given the corresponding local feature \mathbf{a}_k and hyperparameters θ_k . In other words, the hidden variables are conditionally independent on local features, e.g. $p(\xi_{k,i} | \xi_{kj}, \mathbf{a}_k, \theta_k) = p(\xi_{k,i} | \mathbf{a}_k, \theta_k)$. Thus, we can write the following:

$$p(\xi | \mathbf{a}, \Theta) = \prod_k \left(p(\xi_{k,0} | \mathbf{a}_k, \theta_k) p(\xi_{k,00} | \mathbf{a}_k, \theta_k) \cdot \prod_{i \in I_k} p(\xi_{k,i} | \mathbf{a}_k, \theta_k) p(\xi_{k,i0} | \mathbf{a}_k, \theta_k) \prod_{(i,j) \in I_k \times I_k} p(\xi_{k,ij} | \mathbf{a}_k, \theta_k) \right). \tag{19}$$

Since there exist deterministic relationships between any $\xi_{k,\cdot}$ and \mathbf{a}_k , $\xi_{k,\cdot}$ takes value at $f_{k,\cdot}(\mathbf{a}_k; \theta_k)$ with probability 1 given \mathbf{a}_k . Then the conditional probability of $\xi_{k,\cdot}$ on \mathbf{a}_k can be represented with a delta function as in [30]

$$p(\xi_{k,\cdot} | \mathbf{a}_k, \theta_k) = \delta(\xi_{k,\cdot} - f_{k,\cdot}(\mathbf{a}_k; \theta_k)). \tag{20}$$

As a result, $p(\mathbf{u}, \mathbf{h} | \mathbf{a}, \Theta)$ can be expressed with hidden variables ξ included as follows:

$$p(\mathbf{u}, \mathbf{h} | \mathbf{a}, \Theta) = \int p(\mathbf{u}, \mathbf{h} | \xi) p(\xi | \mathbf{a}, \Theta) d\xi. \tag{21}$$

The hidden variables ξ capture fine-scale effects on a coarse-scale. In other words, the influence of high-dimensional stochastic input on responses is represented by ξ . Later on, we will discuss that inference on model responses can be implemented directly on $p(\mathbf{u}, \mathbf{h} | \xi)$ with the information of ξ . Thus without involving the high-dimensional input \mathbf{a} , the computational cost can be significantly reduced. The graphical model with hidden variables is shown in Fig. 3(a).

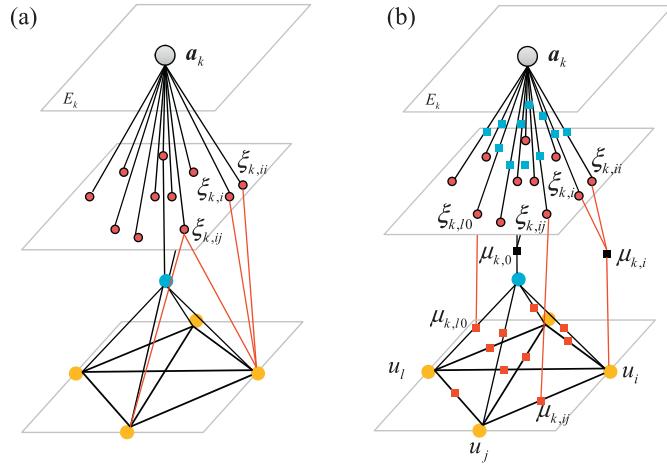


Fig. 3. (a) Undirected graphical model with hidden variables, (b) an equivalent factor graph.

In order to factorize the joint probability $p(\mathbf{u}, \mathbf{h}, \mathbf{a} | \Theta)$ explicitly via the graphical model, the undirected graph with hidden variables in Fig. 3(a) is transformed into a factor graph in Fig. 3(b) in which the factor nodes are potential functions defined through the energy functions in Eq. (13), i.e.

$$\begin{aligned}\mu_{k,i} &:= \exp(-\phi_{k,i}(u_i, \xi_{k,i}, \xi_{k,ii})), \\ \mu_{k,ij} &:= \exp(-\phi_{k,ij}(u_i, u_j, \xi_{k,ij})), \\ \mu_{k,0} &:= \exp(-\phi_{k,0}(h_k, \xi_{k,0}, \xi_{k,00})), \\ \mu_{k,i0} &:= \exp(-\phi_{k,i0}(u_i, h_k, \xi_{k,i0})).\end{aligned}\quad (22)$$

The potential functions for any of the hidden variables ξ and local feature \mathbf{a}_k are simply the conditional distributions (delta functions) in Eq. (20).

Remark 1. The subscripts of feature functions $\phi(\cdot)$ in Eq. (13), potential functions $\mu(\cdot)$ in Eq. (22), hidden variables ξ as well as coefficient functions $f(\cdot)$ are defined according to the following rules: (k, i) denotes the flux component with flux index i , (k, ij) denotes interaction terms between the fluxes u_i and u_j ($i \neq j$), $(k, 0)$ denotes the pressure h_k on the coarse element k and $(k, i0)$ denotes interaction terms between the flux u_i and pressure h_k . All these quantities are defined on the k th coarse element. Note that the index set for flux, I , is a set of positive integers such that $I = \{1, 2, \dots, M_c\}$ where $M_c = \dim(\mathbf{u})$ is the number of model responses for flux. We use 0 to indicate quantities related to pressure.

Remark 2. The hidden variables ξ on different coarse elements k are different. For a stationary permeability random field \mathbf{a} , one can assume that the hyperparameters θ_k are the same on different coarse elements. This is because local features \mathbf{a}_k have the same distribution on coarse elements and one should expect the same relationship between hidden variables and local input on different elements. Thus one can assume that the hidden variables on different elements have the same marginal distribution. However, they cannot be treated as the same variables, $\xi_1 = \xi_2 = \dots = \xi_{N_c}$, as they are associated with local features \mathbf{a}_k . Given a realization of stationary stochastic input $\mathbf{a}^{(t)}$, the local features $\mathbf{a}_k^{(t)}$ and $\mathbf{a}_l^{(t)}$ on elements E_k and E_l are generally different. Consider the realizations of two hidden variables, $\xi_{k,ij}$ and $\xi_{l,pq}$ such that $\xi_{k,ij} = f_{k,ij}(\mathbf{a}_k; \theta_k)$ and $\xi_{l,pq} = f_{l,pq}(\mathbf{a}_l; \theta_l)$. Even though the hyperparameters $\theta_k = \theta_l$, the realizations of the hidden variables are generally different. If we have a nonstationary random field, the hidden variables on different elements are different variables with different marginal distributions.

4. Graphical model parameter learning

We proceed next to learn the various parameters that define the probabilistic graphical model. Suppose we have a training set $\mathcal{D} = \{\mathbf{a}^{(t)}, \mathbf{u}^{(t)}, \mathbf{h}^{(t)}\}_{t=1}^{N_c}$ where $\mathbf{a}^{(t)} = \{\mathbf{a}_k^{(t)}\}_{k=1}^{N_c}$. The likelihood function of training data is formulated in Eq. (12), which can be also written as

$$p(\mathcal{D} | \Theta) = \prod_{t=1}^N p(\mathbf{u}^{(t)}, \mathbf{h}^{(t)} | \mathbf{a}^{(t)}, \Theta) p(\mathbf{a}^{(t)}) \propto \prod_{t=1}^N \exp \left(-\sum_k \mathcal{E}_k(u_{I_k}^{(t)}, h_k^{(t)}; \mathbf{a}_k^{(t)}, \theta_k) \right) \propto \prod_k \exp \left(-\sum_{t=1}^N \mathcal{E}_k(u_{I_k}^{(t)}, h_k^{(t)}; \mathbf{a}_k^{(t)}, \theta_k) \right), \quad (23)$$

where θ_k are constant hyperparameters in element E_k (defined in Eq. (15)). By specifying the prior $p(\Theta)$, the Bayesian posterior of hyperparameters in the probabilistic model is

$$p(\Theta | \mathcal{D}) \propto p(\mathcal{D} | \Theta) p(\Theta). \quad (24)$$

In this work, we set the prior as a multivariate Gaussian distribution with mean zero and an identity covariance matrix. Thus the elements in Θ are mutually independent in the prior, which leads to $p(\Theta) = \prod_k p(\theta_k)$. The posterior distribution in Eq. (24) can then be decomposed as follows:

$$p(\Theta|\mathcal{D}) = \prod_k p(\theta_k|\mathcal{D}) \propto \prod_k \exp\left(-\sum_{t=1}^N \mathcal{E}_k(u_{l_k}^{(t)}, h_k^{(t)}; \mathbf{a}_k^{(t)}, \theta_k)\right) p(\theta_k). \quad (25)$$

As a result, Θ in the probabilistic graphical model can also be estimated locally. We will use a special Monte Carlo method – Sequential Monte Carlo (SMC) [31] to estimate the parameters θ_k on each coarse element through the local posterior distribution $p(\theta_k|\mathcal{D}_k)$, where \mathcal{D}_k are training data related to model responses on element E_k . The parameter values, Θ^* , that maximize the posterior will be taken as fixed parameter values in the probabilistic model to perform probabilistic inference with regard to model responses.

Often posterior distributions can be multi-modal. Conventional MCMC will be trapped by the local modes and long mixing times will be required making it inefficient. In order to explore multi-modal posteriors efficiently, a SMC method is employed [31–35]. First, the idea of annealing/tempering is introduced. Given the target posterior distribution π_n , a sequence of auxiliary distributions, $\{\pi_0, \dots, \pi_n\}$, is proposed to move smoothly from a tractable distribution π_0 to the target π_n . Let the target distribution be the local posterior $p(\theta_k|\mathcal{D}_k)$ in Eq. (25). Then, the following auxiliary distributions are adopted:

$$\pi_t(\theta_k) \propto p^{\gamma_t}(\mathcal{D}_k|\theta_k)p(\theta_k), \quad (26)$$

where $t = 0, 1, \dots, n$ and $0 = \gamma_0 < \gamma_1 < \dots < \gamma_n = 1$ are tempering parameters and $p(\mathcal{D}_k|\theta_k) \propto \exp\left(-\sum_{t=1}^N \mathcal{E}_k(u_{l_k}^{(t)}, h_k^{(t)}; \mathbf{a}_k^{(t)}, \theta_k)\right)$ according to Eq. (25). To simplify the notation, in the following discussion of the SMC algorithm, we use θ to denote the hyperparameters θ_k .

Remark 3. When the stochastic input \mathbf{a} is a stationary random field, the local features have the same joint distribution, $p(\mathbf{a}_1) = p(\mathbf{a}_2) = \dots = p(\mathbf{a}_{N_c})$. In this case, we assume that the hidden variables have the same relationships with the local features in all coarse elements k , i.e. $\Theta \equiv \theta_1 = \theta_2 = \dots = \theta_{N_c}$. Thus the global posterior $p(\Theta|\mathcal{D})$ in Eq. (24) is directly used to infer the hyperparameters. Let $\pi_n \equiv p(\Theta|\mathcal{D})$, then the auxiliary distributions are defined as

$$\pi_t(\Theta) \propto p^{\gamma_t}(\mathcal{D}|\Theta)p(\Theta). \quad (27)$$

The SMC method takes samples from such a sequence of probability distributions based on importance sampling and resampling techniques and constructs a sequential Bayesian inference. At step t , the basic idea is to obtain a large collection of N weighted random samples $\{\theta_t^{(i)}, W_t^{(i)}\}$ ($i = 1, \dots, N$) (also referred to as particles) whose empirical distribution converges asymptotically to the current target distribution π_t . Each particle can be considered as a sample from π_t [36] with an importance weight. According to Eq. (27) or Eq. (26), it is easy to sample directly from π_0 , the prior distribution, at the initial step. The importance sampling technique is performed sequentially to the auxiliary distributions [32]. We move these particles using a predetermined Markov transition kernel [31]. Suppose that, at step $t-1$, we have N samples $\{\theta_{t-1}^{(i)}\}$ distributed in η_{t-1} . A Markov transition kernel K_t with invariant distribution π_t is proposed and new samples are marginally distributed as

$$\eta_t(\theta') = \int \eta_{t-1}(\theta) K_t(\theta, \theta') d\theta. \quad (28)$$

Here, we utilize a Metropolis–Hastings kernel with invariant distribution π_t based on a random walk proposal to move the particles $\theta_{t-1}^{(i)}$ (see Algorithm 1).

An unnormalized importance weight $w_t^{(i)}$ is calculated to estimate the discrepancy between the proposal distribution $\eta_t(\theta)$ and the current auxiliary distribution $\pi_t(\theta)$. A recursive form of the importance weight is

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{\pi_t(\theta_{t-1}^{(i)})}{\pi_{t-1}(\theta_{t-1}^{(i)})}. \quad (29)$$

The normalized weight $W_t^{(i)} = \frac{w_t^{(i)}}{\sum_{k=1}^N w_t^{(k)}}$. More details are given in [31].

It is inevitable that the SMC algorithm will degenerate and the variance of the importance weights would stochastically increase over time [37,38]. Finally only a small portion of particles will have non-negligible weights, which makes the SMC estimate rather inefficient. It is convenient to measure the degeneracy using the effective sample size (ESS) calculated from the normalized importance weights [38]

$$ESS_t = \left(\sum_{i=1}^N (W_t^{(i)})^2 \right)^{-1}. \quad (30)$$

We can define a threshold $ESS_{min} = cN$ ($c < 1$). If $ESS_t < ESS_{min}$, resampling is carried out to relieve the degeneracy of the algorithm. The basic idea is to multiply particles with large weights while discard those with negligible weights. The simplest approach is the multinomial resampling which draws N new samples from $\{\theta_t^{(i)}\}_{i=1:N}$ according to the corresponding normalized weights $\{W_t^{(i)}\}_{i=1:N}$ [31,39]. After resampling, we obtain N weighted particles representing the posterior distribution $\pi_t(\theta)$ at step t . Obviously, the SMC method is directly parallelizable, which further improves the efficiency of the algorithm. The SMC method is summarized in Algorithm 2. Finally, the sample of θ in the particle with the highest weight is taken as the fixed parameter value in the probabilistic model as it corresponds to finding the global maximum of the posterior distribution [31,34].

Algorithm 1. MH kernel with a random walk proposal

1. Sample $u \sim \mathcal{U}(0, 1)$.
 2. Sample $\tilde{\theta} \sim \mathcal{N}(\theta_{t-1}^{(i)}, \sigma_m^2)$.
 3. If $u < \min \left\{ 1, \frac{\pi_t(\tilde{\theta})}{\pi_t(\theta_{t-1}^{(i)})} \right\}$, set $\theta_t^{(i)} = \tilde{\theta}$.
 4. Else set $\theta_t^{(i)} = \theta_{t-1}^{(i)}$.
-

Algorithm 2. SMC algorithm

1. Initialization: For $i = 1, \dots, N$, sample $\theta_0^{(i)} \sim \pi_0(\theta)$ and set the importance weight $W_0(\theta_0^{(i)}) = \frac{1}{N}$.
 2. Updating: At time t , for $i = 1, \dots, N$, sample $\theta_t^{(i)} \sim K_t(\theta_{t-1}^{(i)}, \cdot)$ and set the importance weight according to Eq. (29). Then normalize the importance weight by $W_t^{(i)} = \frac{w_t^{(i)}}{\sum_{k=1}^N w_t^{(k)}}$.
 3. Resampling: Calculate the effective sample size (ESS) by Eq. (30). If $ESS_t < ESS_{min}$, resample the particles $\{\theta_{1:t}^{(i)}, W_t^{(i)}\}$ according to $\{W_t^{(i)}\}$ to obtain a new population $\{\theta_{1:t}^{(i)}, 1/N\}$.
 4. Repeat the above steps until $t = n$, i.e. when the particles are distributed in the last distribution in the sequence.
-

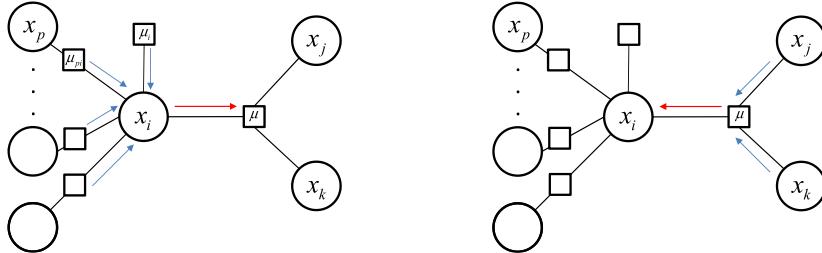


Fig. 4. Message propagation in a factor graph (a) message passing from a variable node to a factor node, (b) message passing from a factor node to a variable node.

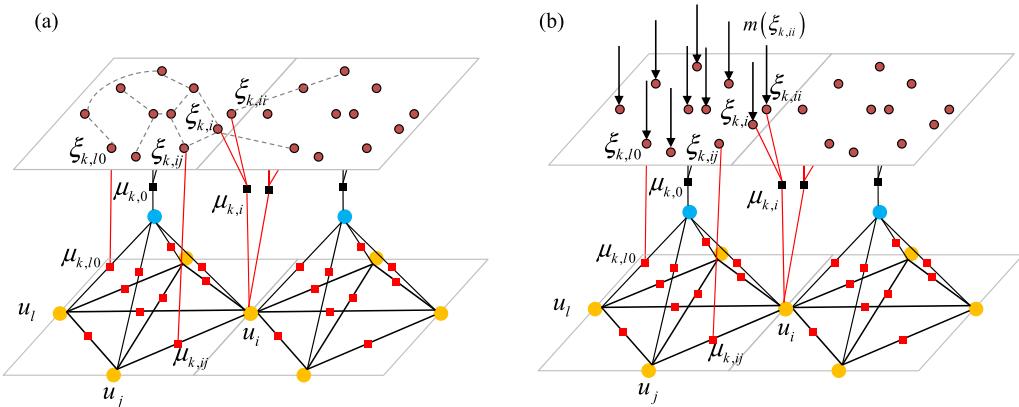


Fig. 5. (a) Reduced factor graph of the probabilistic graphical model in Fig. 3(b) in which the stochastic input α is integrated out, (b) the correlation between hidden variables can be ignored in belief propagation by a direct iterative update of incoming messages $m(\xi_{k,ij})$.

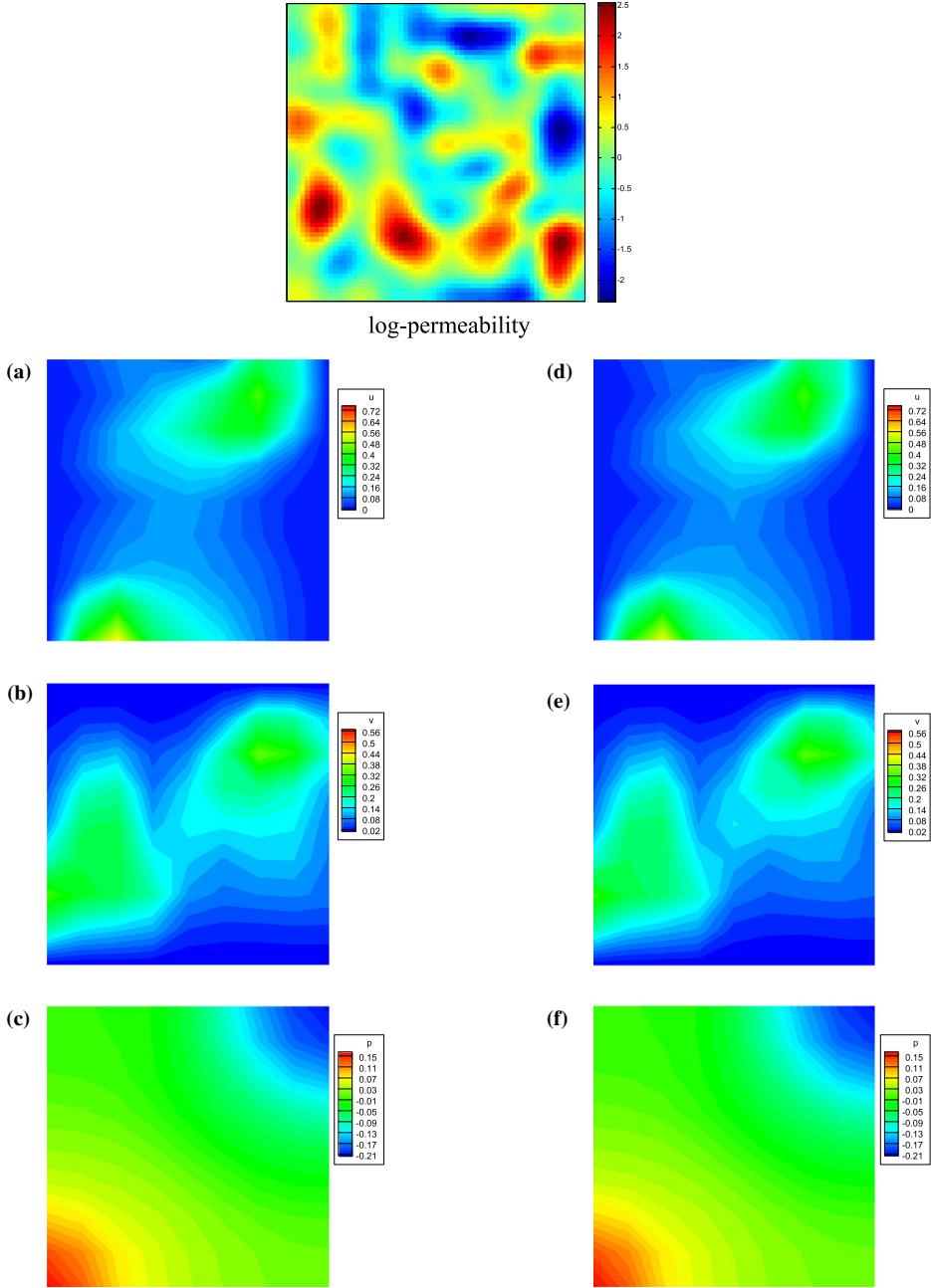


Fig. 6. Isotropic random field: predicted values of model responses given a realization of the stochastic input (a)–(c) x -velocity, y -velocity and pressure obtained from the direct simulation, and (d)–(f) x -velocity, y -velocity and pressure predicted by the probabilistic graphical model (trained with 60 data points).

5. Inference on probabilistic graphical models

The model proposed in Section 3 enables us to perform probabilistic inference with regard to model responses. Suppose the probability distribution of stochastic input $p(\mathbf{a})$ is known, we are then interested in the marginal distributions of responses $p(u_i)$ or $p(h_k)$. This task is challenging as direct marginalization over random variables in the joint distribution is generally intractable. MC methods are extensively used but the convergence rate is slow. Variational methods require discovering good approximating functions [40].

In this work, we will address this problem using belief propagation (BP) – an efficient way of computing marginals of probability distributions from a graphical model. The BP algorithms propagate information through a graphical model via

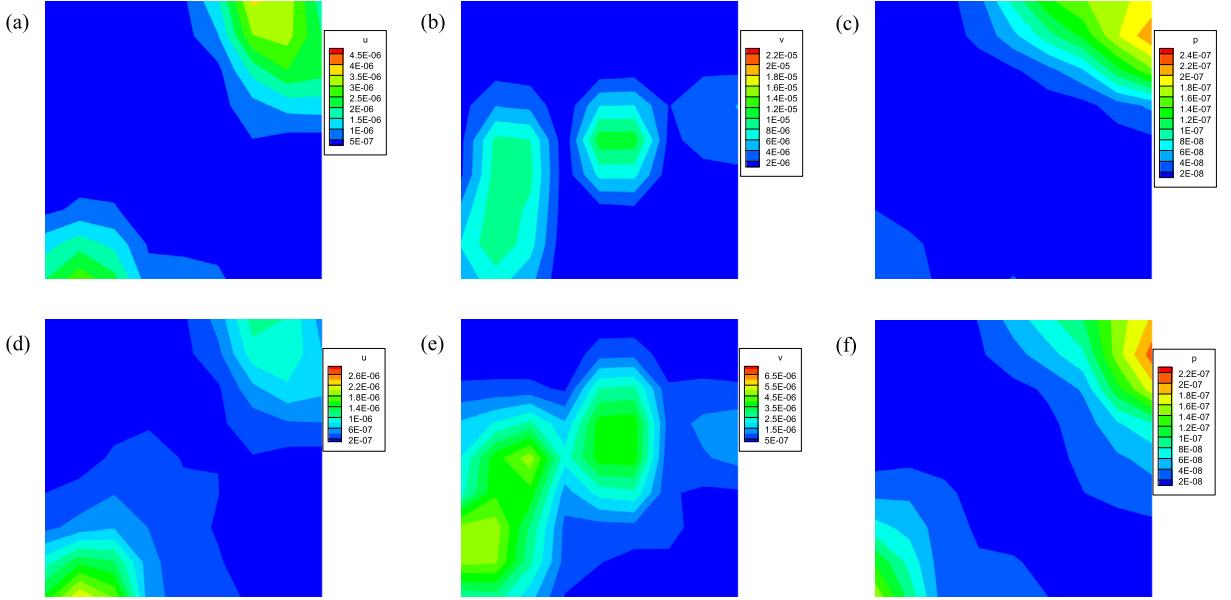


Fig. 7. Isotropic random field: k -fold cross-validation error ($k = 10$) of x -velocity, y -velocity and pressure predicted by the probabilistic graphical model with (a)–(c) 40 samples, and (d)–(f) 60 samples.

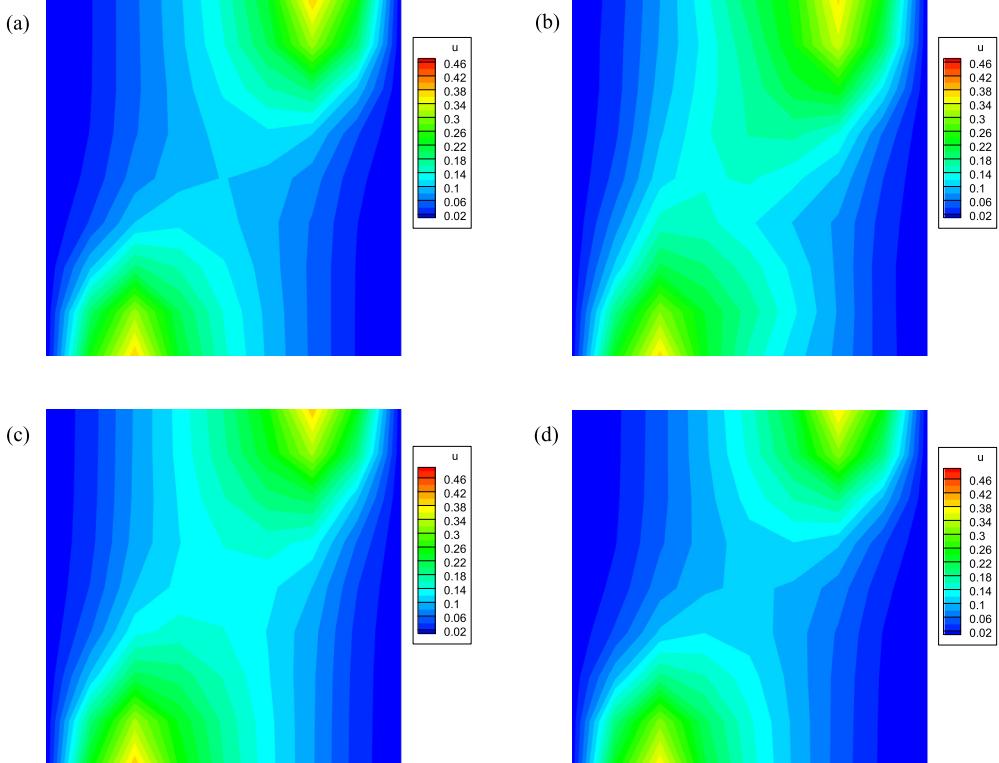


Fig. 8. Isotropic random field: predicted mean of the x -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 20, (c) 40 and (d) 60 data.

messages between neighboring nodes, which is equivalent to applying a local message-passing algorithm [41]. Consider the general factor graph in Fig. 4. Denote all the factor nodes directly linked to variable x_i by $\Gamma(x_i)$. At iteration n of the BP algorithm, the message from x_i to factor node μ is a function of x_i and the update rule is

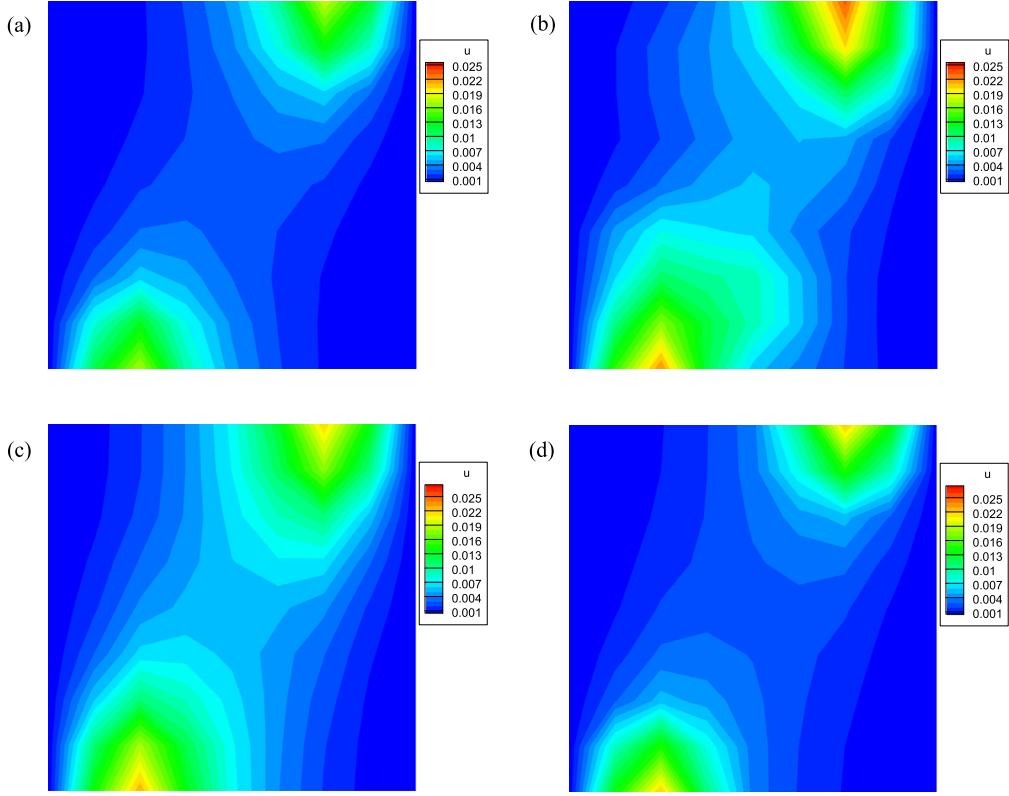


Fig. 9. Isotropic random field: predicted variance of the x -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 20, (c) 40 and (d) 60 data.

$$m_{x_i \rightarrow \mu}^{(n)}(x_i) \leftarrow \mu_i(x_i) \prod_{\mu_{pi} \in \Gamma(x_i) \setminus \mu} m_{\mu_{pi} \rightarrow x_i}^{(n-1)}(x_i). \quad (31)$$

Denote by \mathbf{x}_μ the neighboring variables directly linked to factor node μ , the message from a factor node μ to variable x_i is a function of x_i which is updated by

$$m_{\mu \rightarrow x_i}^{(n)}(x_i) \leftarrow \int_{\mathbf{x}_\mu \setminus x_i} \mu(\mathbf{x}_\mu) \prod_{x_t \in \mathbf{x}_\mu \setminus x_i} m_{x_t \rightarrow \mu}^{(n)}(x_t) d\mathbf{x}_\mu \setminus x_i. \quad (32)$$

When the underlying factor graph is a tree, one iteration of BP is guaranteed to compute the correct posterior marginal distribution of x_i [42]. When the factor graph contains loops, the messages must be updated iteratively until convergence is achieved. An estimate of the posterior marginal distribution of x_i at each iteration is obtained by multiplying all incoming messages from neighboring factor nodes:

$$p^{(n)}(x_i) \propto \mu_i(x_i) \prod_{\mu \in \Gamma(x_i)} m_{\mu \rightarrow x_i}^{(n)}(x_i). \quad (33)$$

Furthermore, this estimation can be extended to joint probability density functions (PDFs). Replacing the variable x_i with a set of correlated nodes \mathbf{x} , the joint PDF can be estimated by multiplying all incoming messages to each element variables in \mathbf{x} with the factor node $\mu(\mathbf{x})$, i.e.

$$p^{(n)}(\mathbf{x}) \propto \mu(\mathbf{x}) \prod_{x_j \in \mathbf{x}} \prod_{\mu \in \Gamma(x_j)} m_{\mu \rightarrow x_j}^{(n)}(x_j). \quad (34)$$

Applying BP algorithms to the factor graph in Fig. 3(b), we can obtain marginal distributions $p(u_i)$ and $p(h_k)$ without generating samples of stochastic input \mathbf{a} or calling the deterministic solver. The challenge lies in computing the message from a factor node to a neighboring variable when fine-scale features \mathbf{a}_k are included in the factor node. In this case, a high-dimensional integration is required according to Eq. (32). In order to solve this problem, the fine-scale features are integrated out of the probabilistic model such that

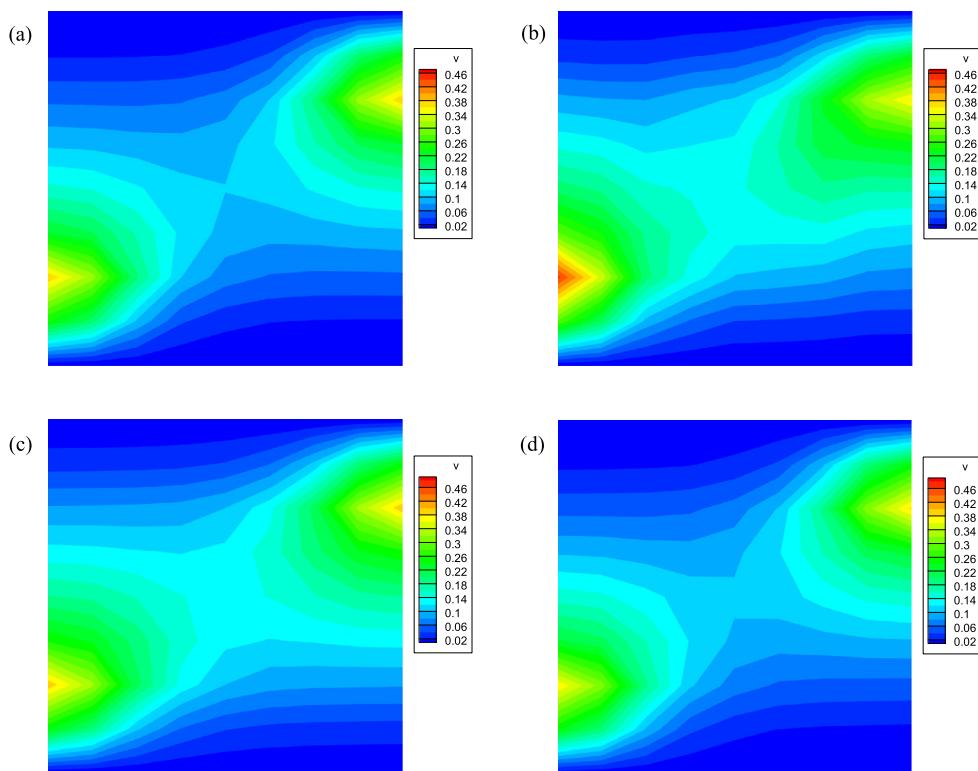


Fig. 10. Isotropic random field: predicted mean of the y -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 20, (c) 40 and (d) 60 data.

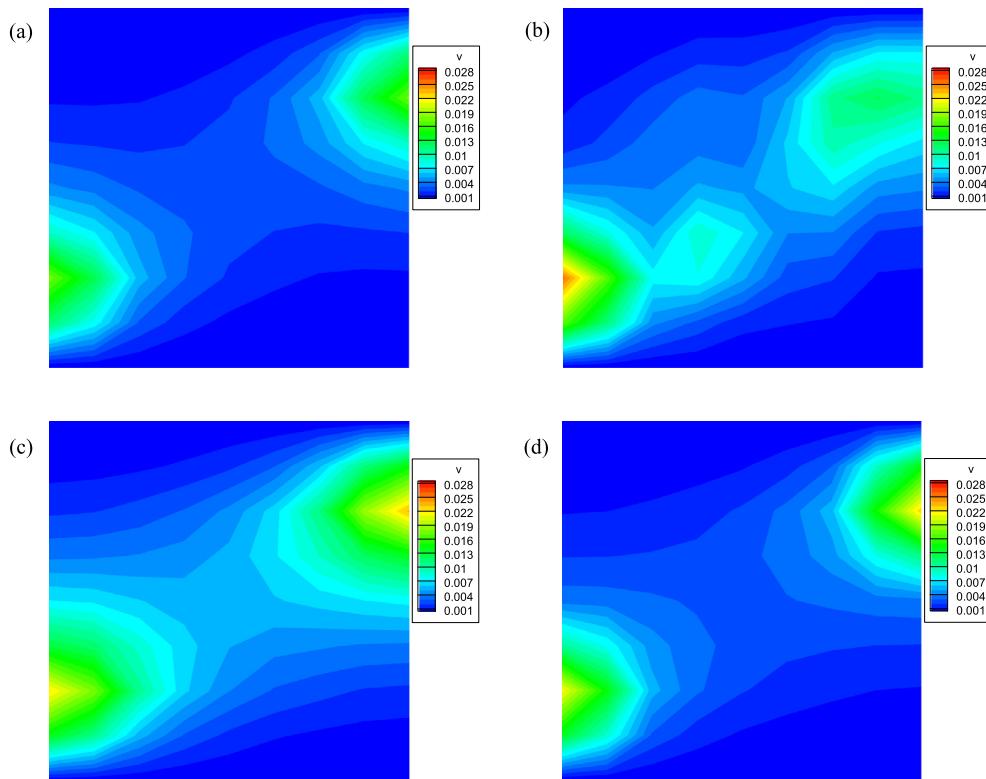


Fig. 11. Isotropic random field: predicted variance of the y -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 20, (c) 40 and (d) 60 data.

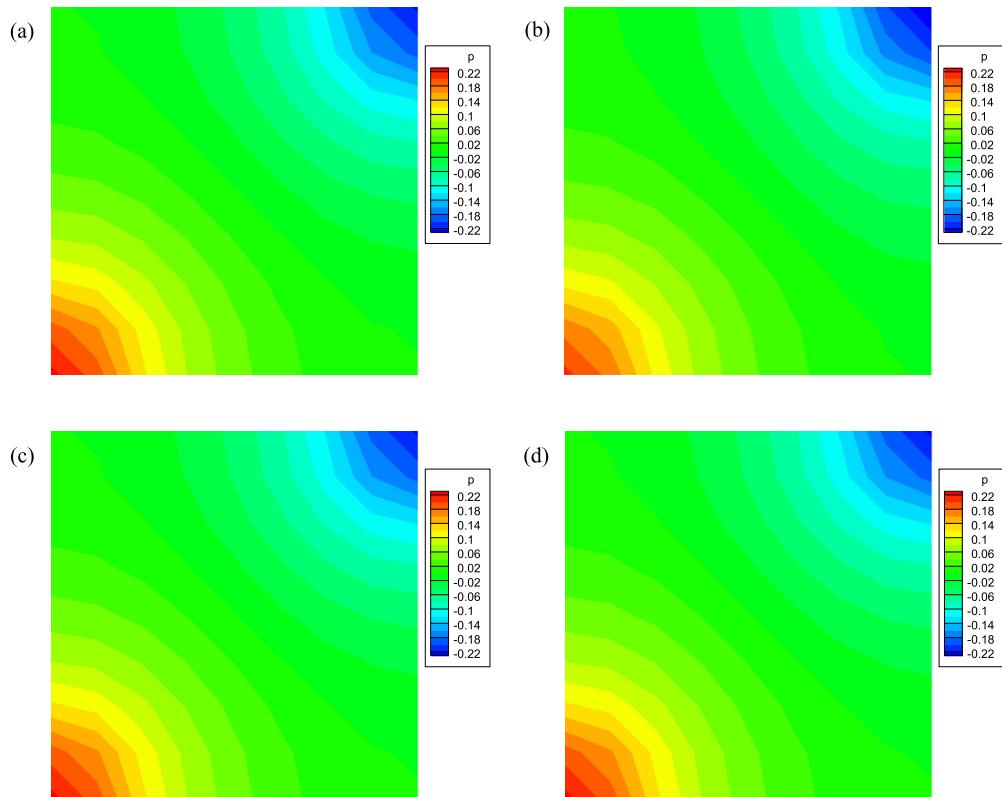


Fig. 12. Isotropic random field: predicted mean of pressure from (a) MC simulation, and from probabilistic graphical models trained by (b) 20, (c) 40 and (d) 60 data.

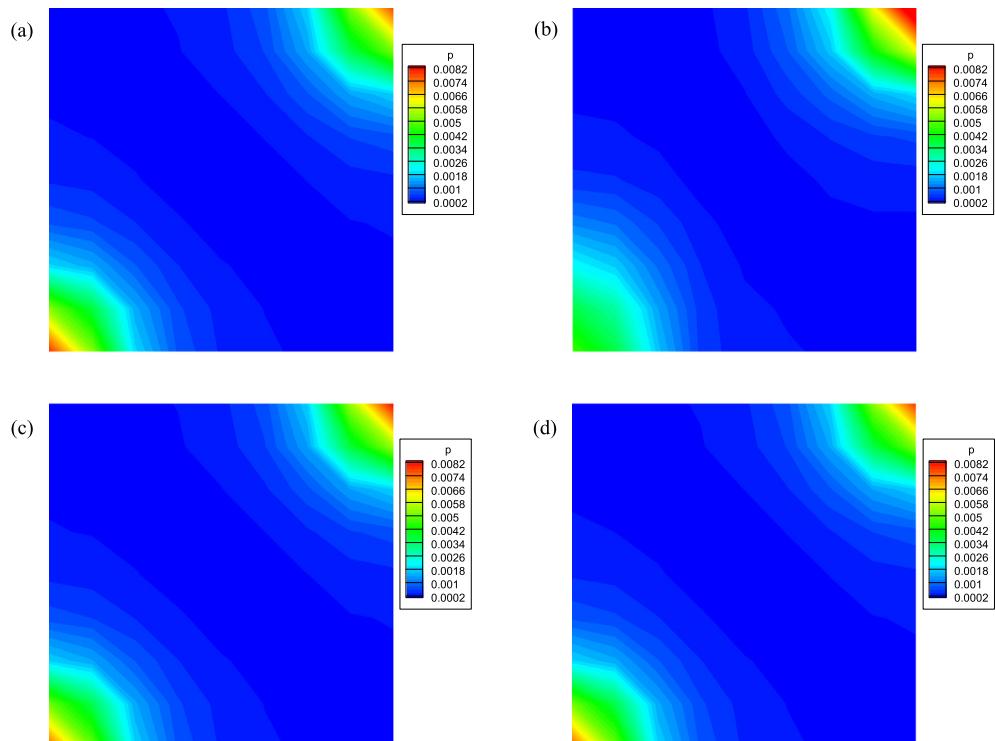


Fig. 13. Isotropic random field: predicted variance of pressure from (a) MC simulation, and from probabilistic graphical models trained by (b) 20, (c) 40 and (d) 60 data.

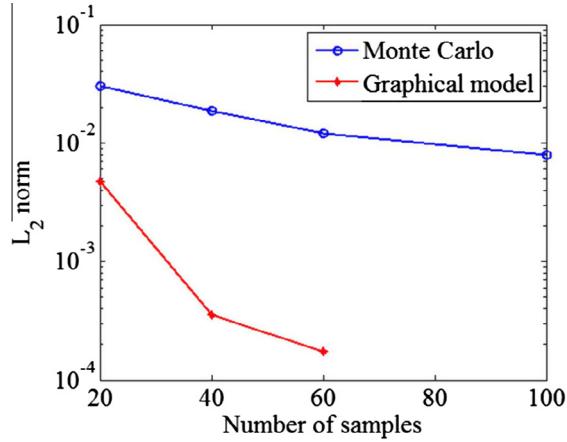


Fig. 14. Isotropic random field: the L_2 norm of the error in the variance of flux as a function of the observed samples for MC simulation and graphical model prediction.

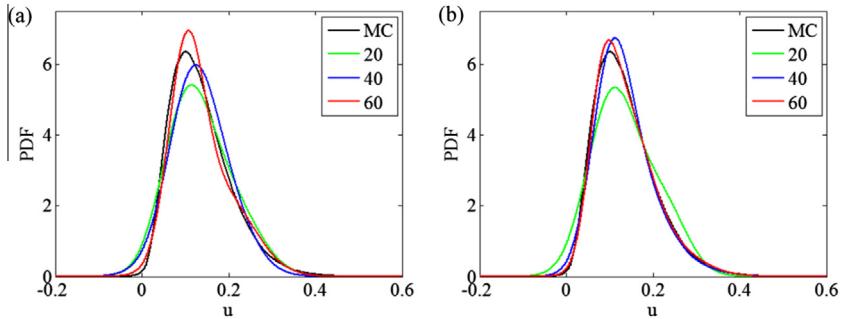


Fig. 15. Isotropic random field: predicted marginal PDF of the x-velocity at point $(0.5, 0.4375)$: Using (a) 2 and (b) 4 Gaussian components in nonparametric messages.

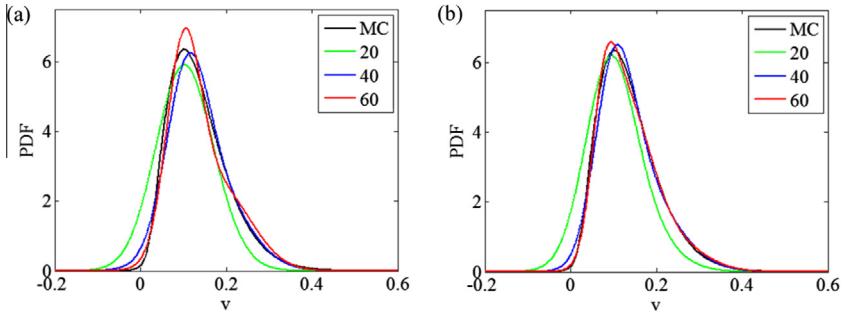


Fig. 16. Isotropic random field: predicted marginal PDF of the y-velocity at point $(0.4375, 0.5)$: Using (a) 2 and (b) 4 Gaussian components in nonparametric messages.

$$p(\mathbf{u}, \mathbf{h}, \xi) = \int p(\mathbf{u}, \mathbf{h} | \xi) p(\xi | \mathbf{a}) p(\mathbf{a}) d\mathbf{a} = p(\mathbf{u}, \mathbf{h} | \xi) p(\xi). \quad (35)$$

Then the factor graph in Fig. 3(b) is transformed to the graph in Fig. 5(a). The hidden variables are correlated in $p(\xi)$ and thus are connected in the graphical model.

In the BP algorithm used in this work, the messages are updated in parallel. At each iteration, we calculate the messages from each factor node to its neighboring variable nodes as well as the messages from each variable node to its neighboring factor nodes based on messages updated in the previous iteration [43]. The messages are considered as converged if their change is less than a threshold in two successive iterations. In the graphical model in Fig. 5(a), there exists a unique message between any factor node (potential function) and any of its arguments, including (1) messages between factor node $\mu_{k,ij}$ and

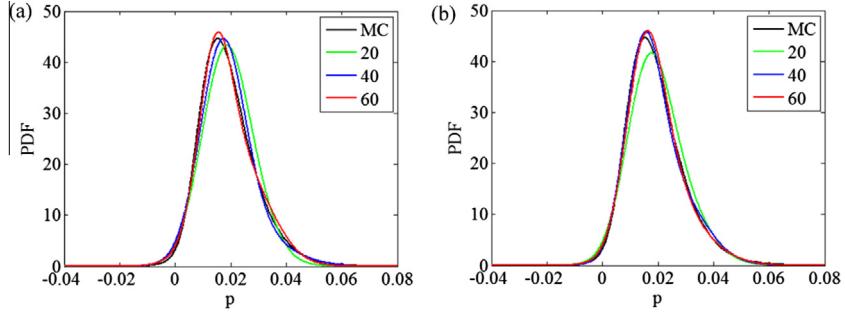


Fig. 17. Isotropic random field: predicted marginal PDF of pressure at the coarse element centered at point (0.4375, 0.4375): Using (a) 2 and (b) 4 Gaussian components in nonparametric messages.

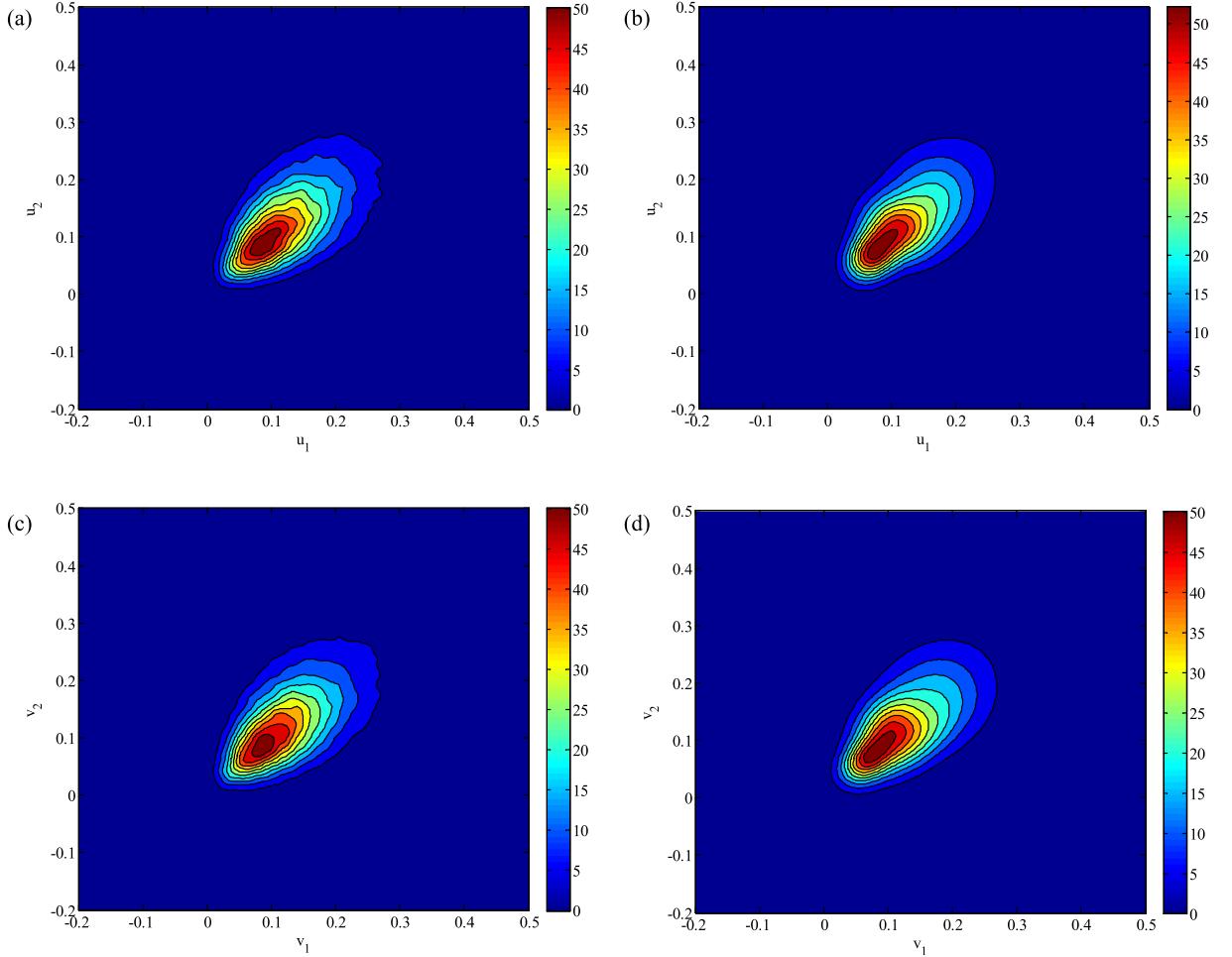


Fig. 18. Isotropic random field: the joint PDF of the x -velocity u_1 at (0.5, 0.4375) and u_2 at (0.375, 0.4375); (a) direct simulation (b) probabilistic graphical model; the joint PDF of y -velocity v_1 at (0.4375, 0.5) and v_2 at (0.4375, 0.375): (c) direct simulation (d) probabilistic graphical model.

any of its arguments, u_i, u_j or $\xi_{k,ij}$; (2) messages between factor node $\mu_{kk,l}$ and any of its arguments, u_i, h_k or $\xi_{kk,l}$; (3) messages between factor node $\mu_{k,i}$ and any of its arguments, $u_i, \xi_{k,i}$ or $\xi_{k,ii}$; (4) messages between factor node μ_{kk} and any of its arguments, h_k, ξ_{kk} or ξ_{kkk} ; (5) messages between hidden variables. The messages in (5) are more complex and will be discussed separately later on. The messages in (1)–(4), since there is no prior information, are represented nonparametrically as weighted Gaussian mixtures. Without loss of generality, consider the message from factor node $\mu_{k,ij}(u_i, u_j, \xi_{k,ij})$ to variable node u_i , which is approximated by

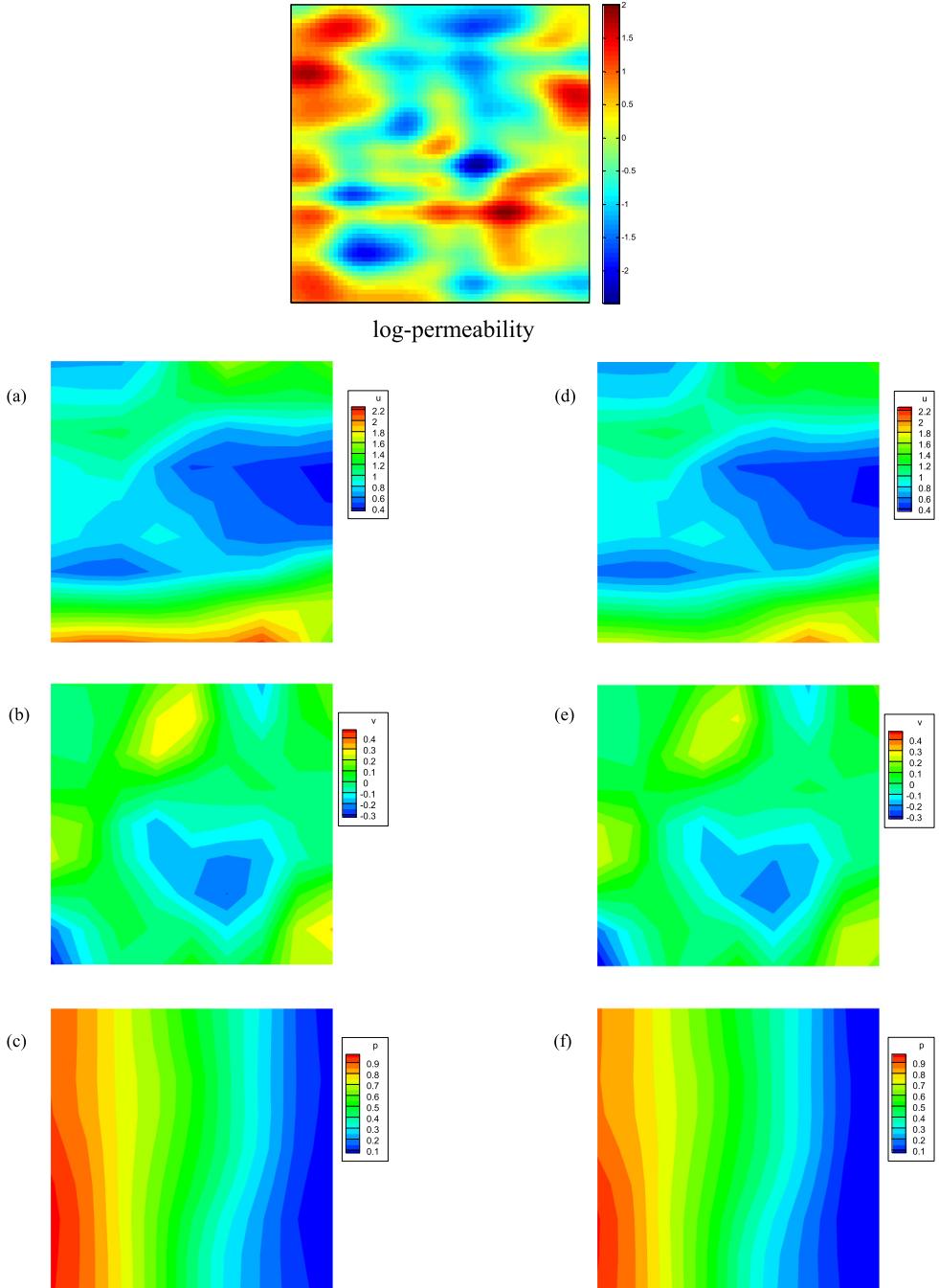


Fig. 19. Anisotropic random field: predicted values of model responses given a realization of the stochastic input (a)–(c) x -velocity, y -velocity and pressure obtained from direct simulation, and (d)–(f) x -velocity, y -velocity and pressure predicted by the probabilistic graphical model (trained with 2400 data points).

$$m_{\mu_{k,j} \rightarrow u_i}(u_i) \approx \sum_{t=1}^T l_t \mathcal{N}(u_i; \bar{u}_i^t, \sigma_i^2), \quad (36)$$

where l_t is the weight of a Gaussian kernel with mean \bar{u}_i^t and variance σ_i^2 [44]. At iteration n of the BP algorithm, the messages between factor nodes and variables nodes are updated according to Eqs. (31) and (32), i.e.

$$m_{\mu_{k,j} \rightarrow u_i}^{(n)}(u_i) \leftarrow \int \mu_{k,j}(u_i, u_j, \xi_{k,j}) m_{u_j \rightarrow \mu_{k,j}}^{(n)}(u_j) m_{\xi_{k,j} \rightarrow \mu_{k,j}}^{(n)}(\xi_{k,j}) d\xi_{k,j} du_j, \quad (37)$$

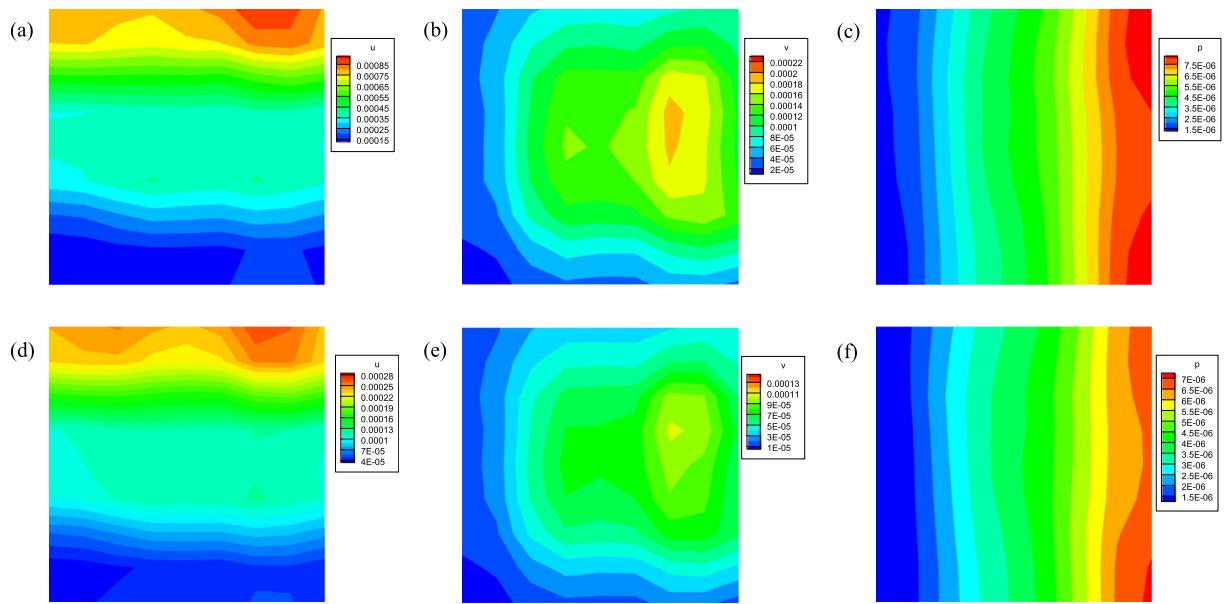


Fig. 20. Anisotropic random field: k -fold cross-validation error ($k = 10$) of x -velocity, y -velocity and pressure predicted by the probabilistic graphical model with (a)–(c) 1600 samples, and (d)–(f) 2400 samples.

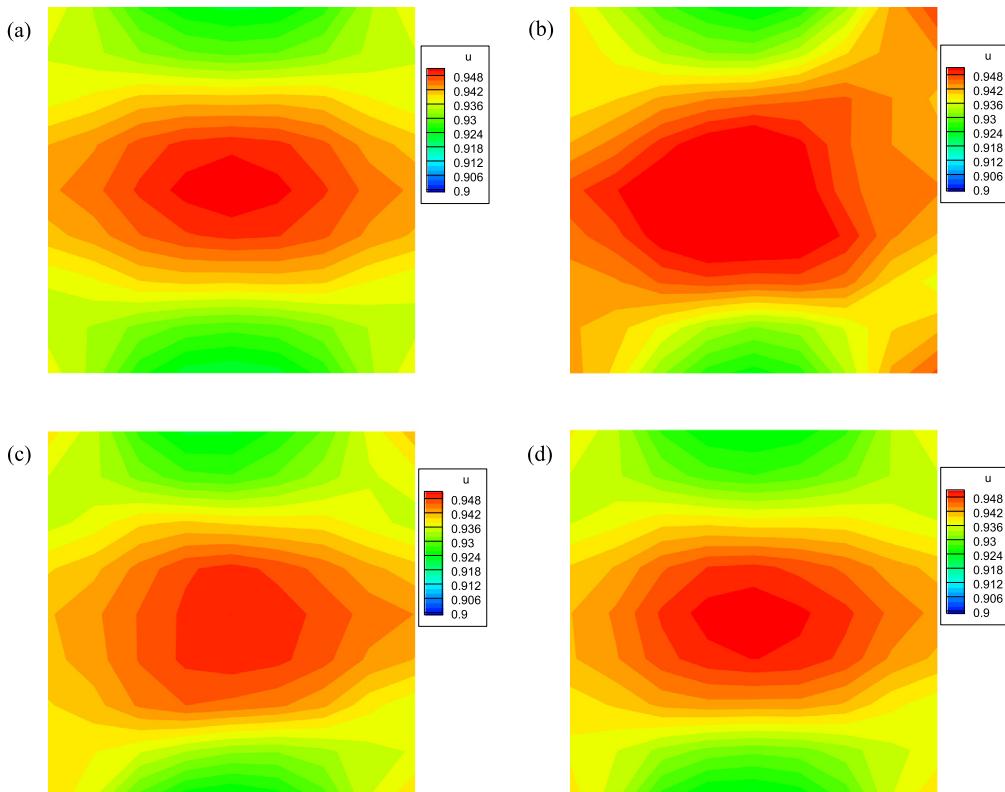


Fig. 21. Anisotropic random field: predicted mean of the x -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

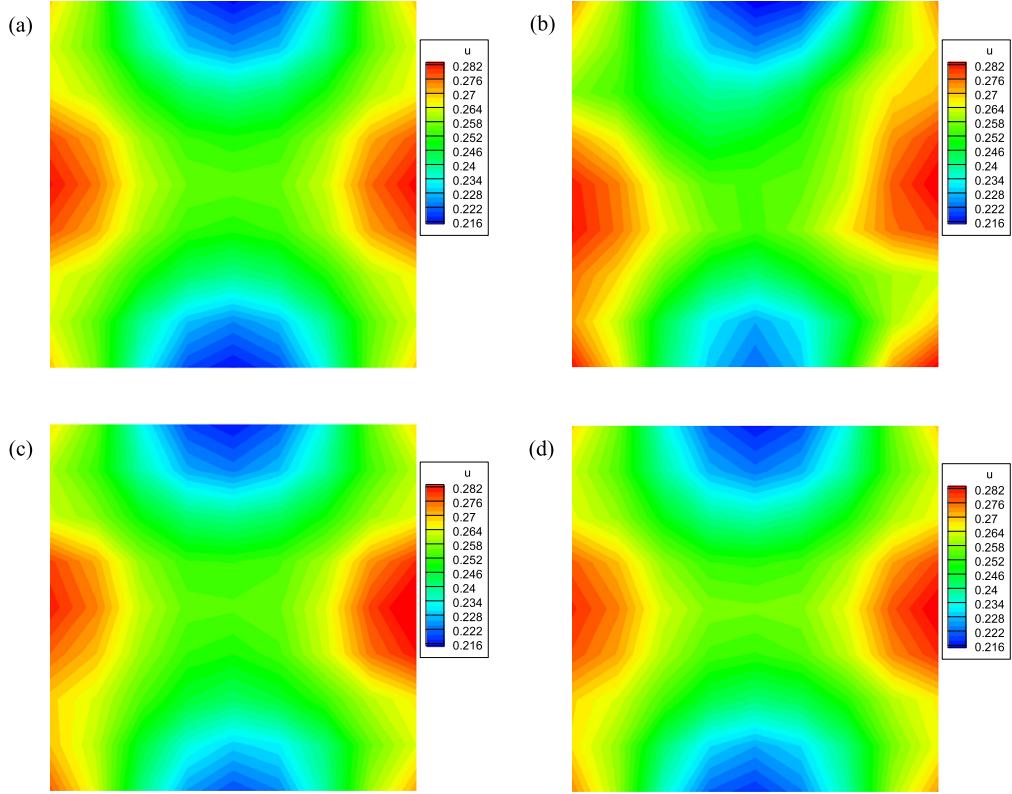


Fig. 22. Anisotropic random field: predicted variance of the x -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

and the message $m_{u_i \rightarrow \mu_{k,ij}}^{(n)}(u_i)$ is simply the product of all incoming messages from neighboring factor nodes to u_i at iteration $n - 1$. However, as discussed in [44], a BP update which multiplies d Gaussian mixtures, each containing T components can produce a Gaussian mixture with T^d components, i.e. the number of mixture components increases exponentially. Therefore, when updating the message $m_{\mu_{k,ij} \rightarrow u_i}(u_i)$ in Eq. (37), we draw samples $(u_i, u_j, \xi_{k,ij})$ from $\mu_{k,ij}(u_i, u_j, \xi_{k,ij}) \cdot m_{u_j \rightarrow \mu_{k,ij}}^{(n)}(u_j) \cdot m_{\xi_{k,ij} \rightarrow \mu_{k,ij}}^{(n)}(\xi_{k,ij})$ using MCMC. Then the message is approximated using a Gaussian mixture model with T kernels as in Eq. (36) from samples of u_i [45].

The main remaining challenge lies in the update of messages between the hidden variables ξ . Although analytic expressions of $p(\mathbf{a})$ and $p(\xi|\mathbf{a})$ are explicit, the joint distribution of hidden variables ξ could be complicated such that the links between them are implicit when stochastic input has been removed from the graphical model. To bypass the difficulties in passing messages between hidden variables, we examine the four messages related to a hidden variable $\xi_{k,ij}$, i.e. (1) the message from a factor node $\mu_{k,ij}$, denoted by $m_{\mu_{k,ij} \rightarrow \xi_{k,ij}}$, (2) the message from $\xi_{k,ij}$ to factor node $\mu_{k,ij}$, (3) the message from other hidden variables, denoted by $m(\xi_{k,ij})$, and (4) the messages from $\xi_{k,ij}$ to other hidden variables. According to Fig. 5(a) and Eq. (31), the message in (2) equals to the message in (3). On the other hand, according to Eq. (33), the messages in (1) and (3) are correlated in the following way

$$p(\xi_{k,ij}) \propto m(\xi_{k,ij}) m_{\mu_{k,ij} \rightarrow \xi_{k,ij}}^{(n)}(\xi_{k,ij}). \quad (38)$$

The marginal distributions of hidden variables, $p(\xi_{k,\cdot})$ can be obtained by standard kernel density estimators given samples of \mathbf{a}_k (see Eq. (15)) and are thus known. As a result, the messages between hidden variables are updated directly via $p(\xi_{k,ij})$, i.e. at iteration n , the input messages $m(\xi_{k,ij})$ are updated by

$$m^{(n+1)}(\xi_{k,ij}) \propto p(\xi_{k,ij}) / m_{\mu_{k,ij} \rightarrow \xi_{k,ij}}^{(n)}(\xi_{k,ij}). \quad (39)$$

Then the only messages undetermined are those from $\xi_{k,ij}$ to other hidden variables. In theory, they are used to compute the incoming messages to other hidden variables, which, however, can be estimated in the same way as in Eq. (39). Therefore, the messages between hidden variables do not play any role in belief propagation and the graphical model in Fig. 5(a) is

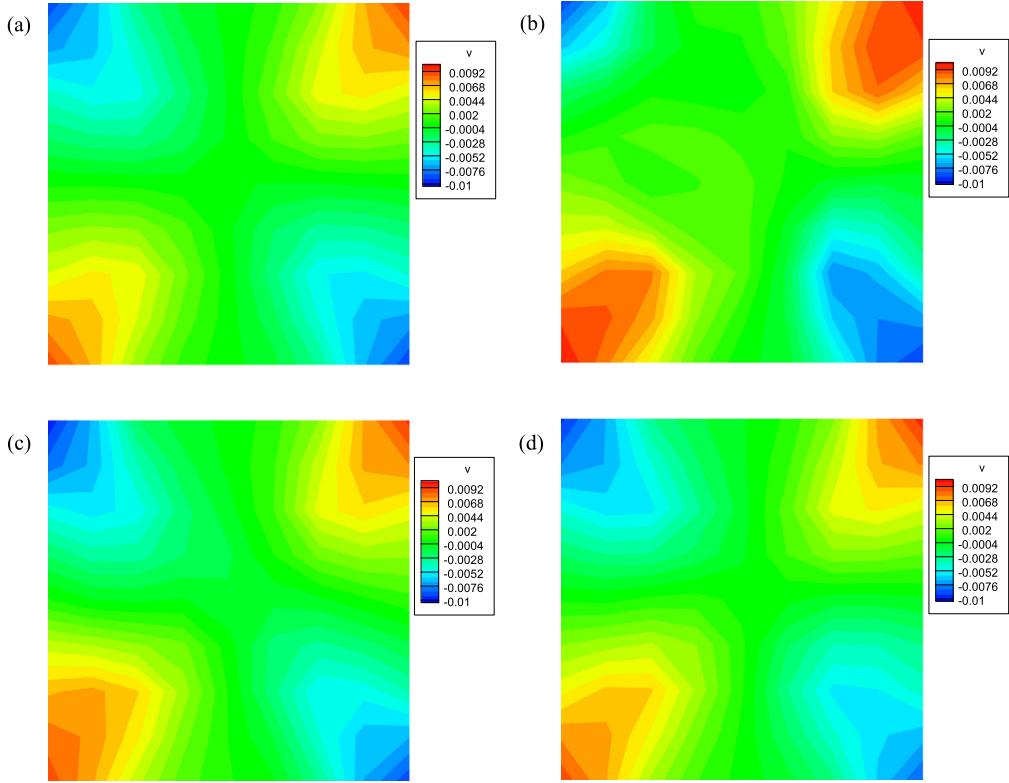


Fig. 23. Anisotropic random field: predicted mean of the y -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

transformed into the one in Fig. 5(b). Finally, starting from initial guess of all messages, the BP algorithm iteratively updates the messages until the marginal distributions of responses $\{u_i\}$ and $\{h_k\}$ converge. The procedure is summarized in Algorithm 3.

Algorithm 3. A general Belief Propagation with nonparametric messages

1. Initialization: Set initial input messages of hidden variables as their marginal distributions obtained from sampling, i.e. $m^0(\xi_{k,:}) = p(\xi_{k,:})$ and all the other messages as Gaussian mixtures defined in Eq. (36).
2. Iterate: At step n , update messages according to Eqs. (37) and (39).
3. Convergence: The marginal distributions $p(u_i)$ (and also for $p(h_k)$) are approximated by

$$p(u_i) \propto \prod_{\mu \in \Gamma(u_i)} m_{\mu \rightarrow i}(u_i) \approx \sum_{t=1}^T l_t \mathcal{N}(u_i; \bar{u}_i^t, \sigma_i^2),$$

from the sampling-based method. The marginal distributions $\{p(u_i)\}$ converge when $\max \| \bar{u}_i^t - \bar{u}_i^{t-1} \| < \epsilon$. Stop iteration when all marginal distributions converge.

Remark 4. Given a realization of the stochastic input, $\mathbf{a}^{(n)}$, the values of the hidden variables, e.g. $\xi_{k,ij}^{(n)}$, can be directly obtained through the function $f_{k,ij}$, i.e. $\xi_{k,ij}^{(n)} = f_{k,ij}(\mathbf{a}_k^{(n)})$. As the hidden variables in Fig. 5 are observed, there is no message $m(\xi_{:,i})$ between them. Then the factor graph in Fig. 5(b) corresponds to the conditional distribution $p(\mathbf{u}, \mathbf{h} | \xi^{(n)})$. The unobserved variables in this graph are the model responses (\mathbf{u}, \mathbf{h}) . When belief propagation is performed, we obtain the marginal distributions of the model responses conditioned on the input, e.g. $p(u_i | \mathbf{a}^{(n)})$ and $p(h_k | \mathbf{a}^{(n)})$. As a result, let the expectations $\mathbb{E}(u_i | \mathbf{a}^{(n)})$ and $\mathbb{E}(h_k | \mathbf{a}^{(n)})$ be the predicted values of model responses. We can then obtain a surrogate model by running the belief propagation algorithm on a factor graph given a realization of the stochastic input.

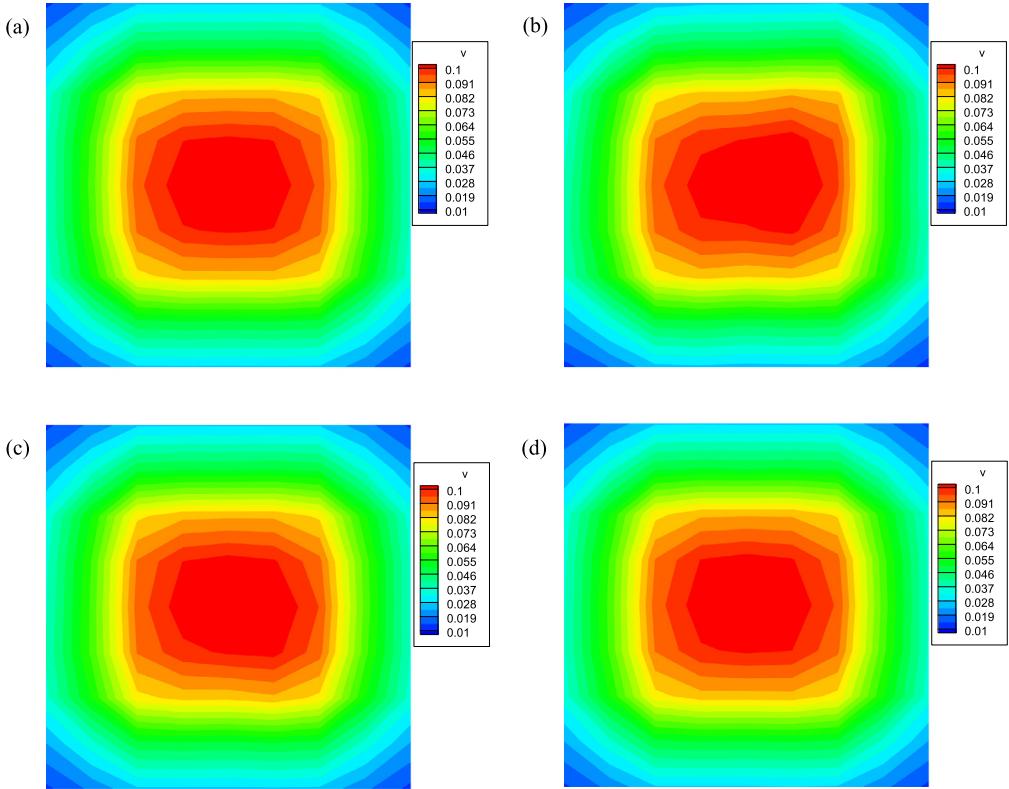


Fig. 24. Anisotropic random field: predicted variance of the y -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

6. Numerical examples

In this section, we construct probabilistic graphical models to predict fluid flow in random heterogeneous porous media. The domain is a unit square $[0, 1]^2$. The permeability is defined on a 64×64 fine grid and we are interested in the flux at the middle point of edges of coarse elements as well as the pressure on a 8×8 coarse grid. The model responses in the training data, $\mathcal{D} = \{\mathbf{a}^{(i)}, \mathbf{h}^{(i)}, \mathbf{u}^{(i)}\}$, are generated using a mixed finite element method on the fine grid and are collected on locations related to the coarse grid [46,47]. We choose $r = 4$ kernels in Eq. (15) to approximate the relationship between the hidden variables and the local features. Since there are 20 hidden variables on each coarse element, there are totally 80 hyperparameters in θ_k associated with each coarse element. In SMC learning of these hyperparameters, we choose standard Gaussian distribution $\mathcal{N}(0, 1)$ as the prior for each component of hyperparameters. The threshold of ESS is set to be $\text{ESS}_{\min} = 0.85N$. A linear cooling schedule is selected for γ_t in Eq. (27). For 500 time steps, the sequence $\{\gamma_0, \dots, \gamma_{500}\}$ increases uniformly from 0 to 1. In SMC, we employ 800 particles.

6.1. Isotropic random field

In this example, the log-permeability \mathbf{a} is a Gaussian random field with mean zero and an exponential covariance function defined as

$$\text{cov}(\mathbf{x}, \mathbf{x}^*) = \sigma^2 \exp\left(-\frac{|x_1 - x_1^*|}{L_1} - \frac{|x_2 - x_2^*|}{L_2}\right), \quad (40)$$

where coordinates $\mathbf{x} = (x_1, x_2)$ and $\mathbf{x}^* = (x_1^*, x_2^*)$ and σ is the standard deviation of the random field. An isotropic random field is assumed such that correlation lengths $L_1 = L_2 = 0.1$ and the standard deviation $\sigma = 1.0$. The samples of permeability are generated using standard Karhunen–Loëve (KL) expansion with the first 100 terms. Since this is a stationary random field, the local features \mathbf{a}_k are subject to the same distribution. As hidden variables capture local features on each coarse element, it is reasonable to assume the same relationships between the hidden variables and the local features on different coarse elements. Thus the hyperparameters defined in Eq. (15) on coarse elements are identical, i.e. $\theta = \theta_1 = \dots = \theta_{N_c}$ and $\Theta \equiv \theta$ in Eq. (23). The global posterior distribution, $p(\Theta|\mathcal{D})$, defined in Eq. (23) is directly used to estimate the hyperparameters Θ . In this example, $N = 20, 40, 60$ training data are generated to train the probabilistic model.

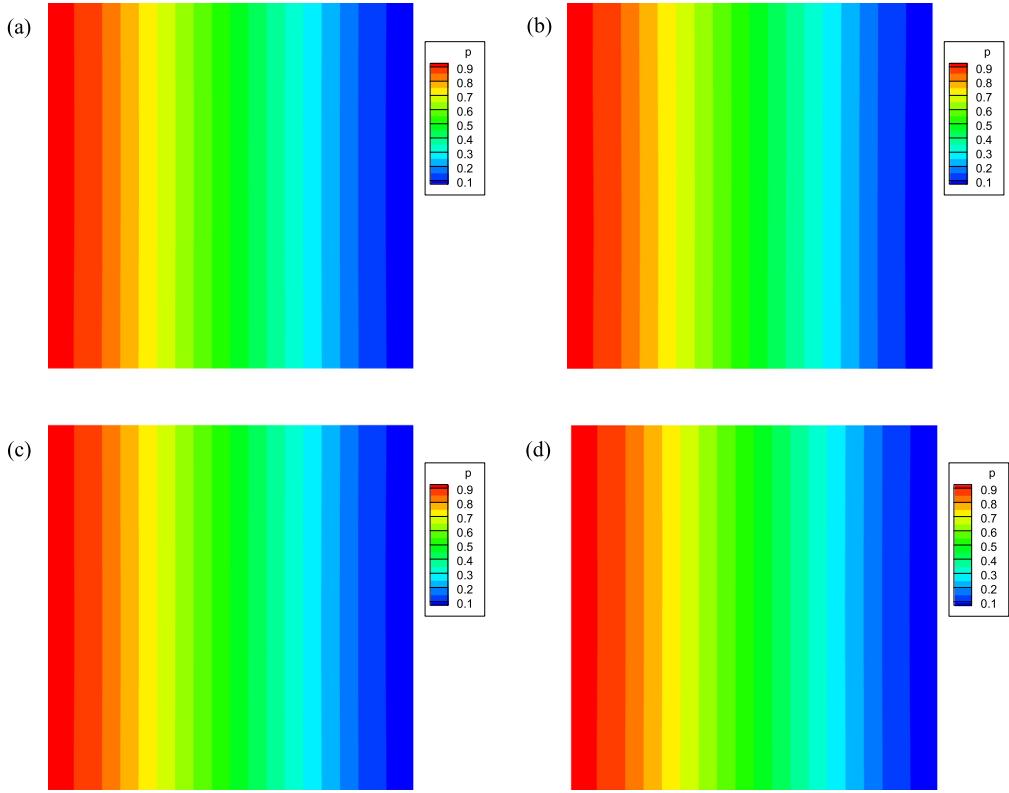


Fig. 25. Anisotropic random field: predicted mean of pressure from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

The source term f in Eq. (3) is used to model injection/production wells:

$$f(\mathbf{x}) = \begin{cases} -r, & \text{if } 0 \leq x_i < w, \text{ for } i = 1, 2, \\ r, & \text{if } 1 - w \leq x_i < 1, \text{ for } i = 1, 2, \\ 0 & \text{otherwise.} \end{cases} \quad (41)$$

The parameters are chosen to be $r = 10$ and $w = 1/8$. No-flow homogeneous Neumann boundary conditions are applied on all boundaries. The threshold for BP convergence is set to $\epsilon = 10^{-4}$. Note that the reference solutions (mean, variance and marginal PDFs) are obtained by MC simulation with 10^6 samples.

According to the belief propagation algorithm proposed in Section 5, the estimated marginal distributions are Gaussian mixtures (or Gaussian if there is only one Gaussian component in each message). In order to verify the correctness of probabilistic graphical models, we randomly generate a realization of stochastic input and predict the model responses using belief propagation as discussed in Remark 4. The messages in belief propagation are assumed to be Gaussian functions. The graphical model is trained with 60 data points. In comparison with the direct simulation results obtained from the mixed multiscale FEM (Fig. 6), we can see that accurate predictions are obtained from the probabilistic graphical model given an observation of stochastic input. To quantitatively estimate the predictive accuracy of the probabilistic graphical model, a k -fold ($k = 10$) cross-validation is applied [48]. It is performed with 40 and 60 samples, respectively (Fig. 7). On each fold, the mean squared prediction error for each element of model responses is obtained and the average of errors is taken on the k folds.

Next, we use belief propagation to estimate the statistics of the model responses without given realizations of input, i.e. all variables in Fig. 5 are unobserved. Four Gaussian mixture components are adopted in the messages in belief propagation. The predicted mean and variance of the model responses (velocities and pressure) are compared with the MC solutions in Figs. 8–13. The comparison shows that more training data can generate probabilistic models with higher predictive accuracy. The convergence plot in Fig. 14 shows that the error in the variance of the flux predicted by the probabilistic graphical model decreases much faster than that in the standard MC simulation. The reference solution here is taken as Monte Carlo with 10^6 samples. With a stationary random field as stochastic input, the hyperparameters do not vary on coarse elements. As a result, they can be accurately estimated with a small number of data sets. However, the same number of samples is far from sufficient for convergence in the MC simulation.

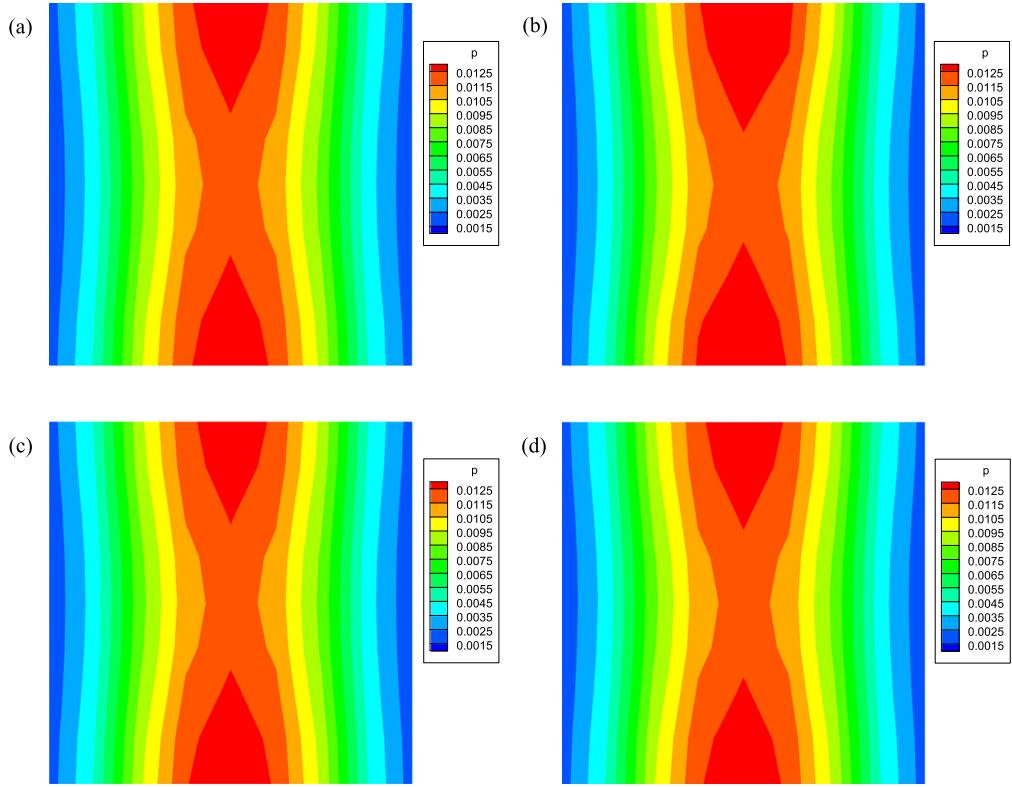


Fig. 26. Anisotropic random field: predicted variance of pressure from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

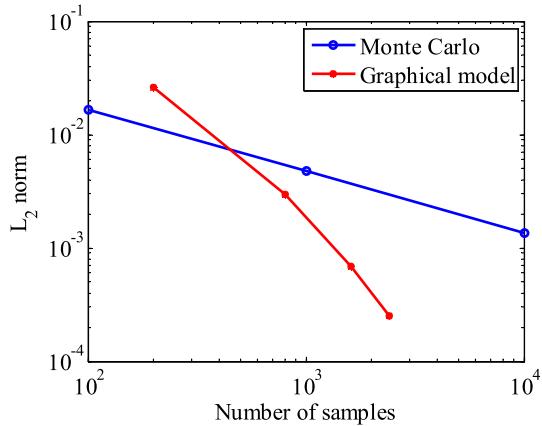


Fig. 27. Anisotropic random field: the L_2 norm of the error in the variance of flux as a function of the observed samples for MC simulation and graphical model prediction.

In the belief propagation algorithm, the number of Gaussian components in the nonparametric messages should be specified. In this example, we set the number of components as 2 and 4 and predict the marginal distributions for each case. Fig. 15 shows the predicted PDFs of the x -velocity at point $(0.5, 0.4375)$ in the spatial domain. Fig. 16 shows the predicted PDFs of y -velocity at point $(0.4375, 0.5)$ and Fig. 17 shows the predicted PDFs of pressure on a coarse element centered at $(0.4375, 0.4375)$. Obviously, when the target marginal PDF obtained from MC simulation is non-Gaussian, the assumption of Gaussian messages cannot apply. With sufficient training data, belief propagation with 4 components in the messages generally gives better prediction of the marginal PDF than that with only 2 components in the messages. In addition to the first-order marginal PDFs, the probabilistic graphical model can also capture the correlations of model responses.

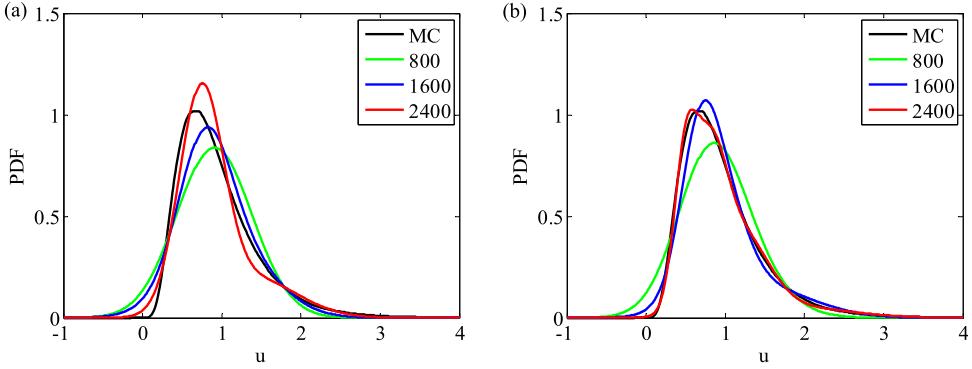


Fig. 28. Anisotropic random field: predicted marginal PDF of the x -velocity at point $(0.5, 0.4375)$: Using (a) 2 and (b) 4 Gaussian components in nonparametric messages.

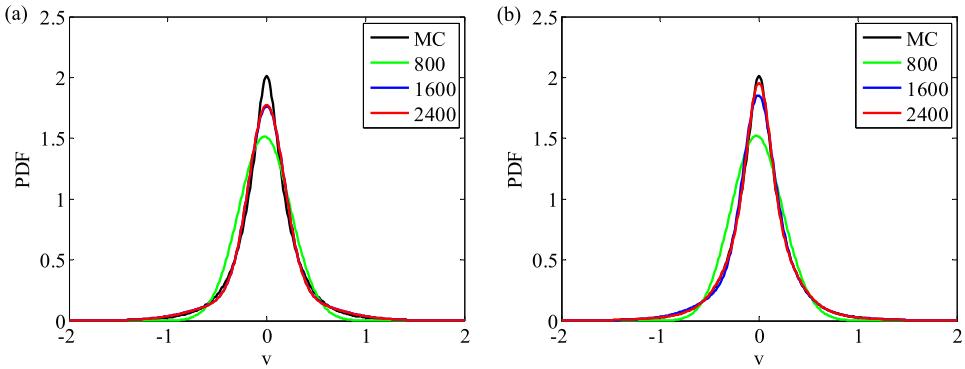


Fig. 29. Anisotropic random field: predicted marginal PDF of the y -velocity at point $(0.4375, 0.5)$: (a) 2 and (b) 4 Gaussian components in nonparametric messages.

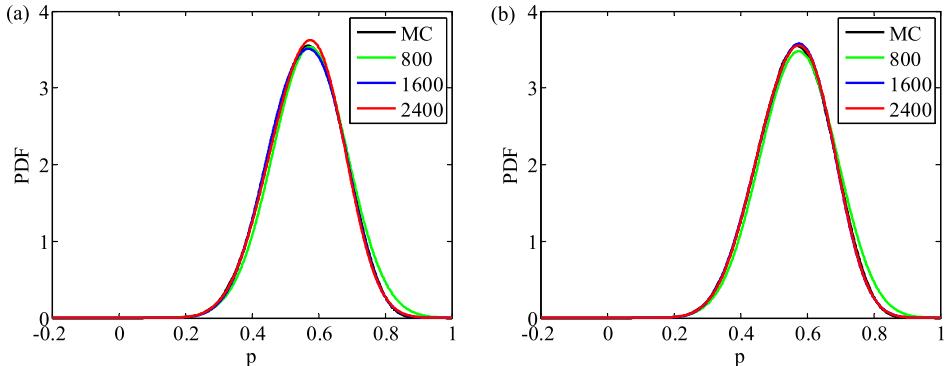


Fig. 30. Anisotropic random field: predicted marginal PDF of pressure at the coarse element centered at point $(0.4375, 0.4375)$: (a) 2 and (b) 4 Gaussian components in nonparametric messages.

Replacing the variable in Eq. (33) with a pair of connected nodes, the joint PDF can be estimated by multiplying all incoming messages to both nodes with the factor node between them. In Fig. 18, the joint distributions of x -velocity and y -velocity at two different locations are presented. Compared with results from direct simulations, the non-Gaussian joint distributions are accurately estimated with a probabilistic graphical model trained by 60 data points.

6.2. Anisotropic random field

In this example, an anisotropic Gaussian random field α is assumed with correlation length $L_1 = 0.1$, $L_2 = 0.2$ and standard deviation $\sigma = 1.0$. The samples of log-permeability are also generated using KL expansion with the first 100 terms. As

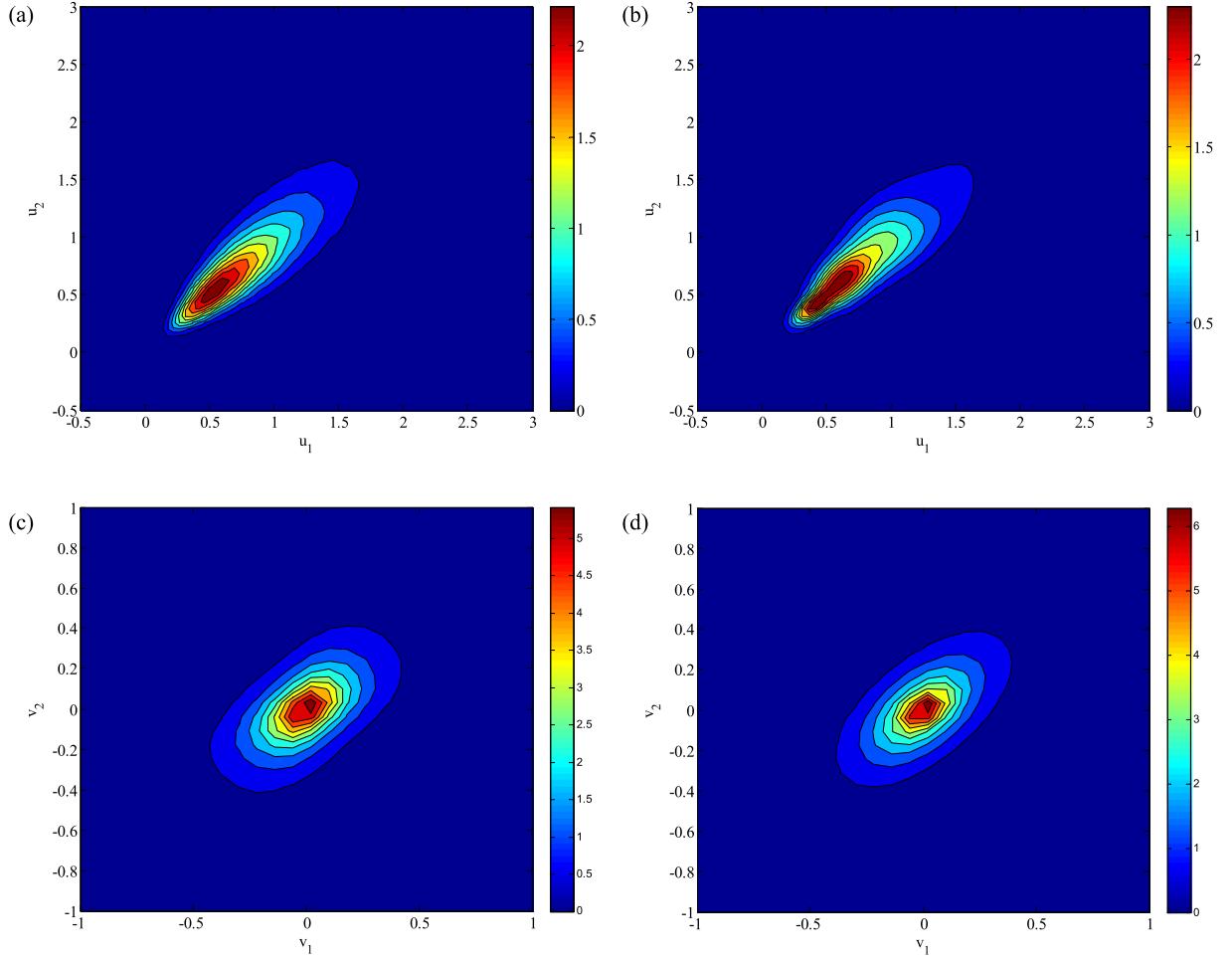


Fig. 31. Anisotropic random field: the joint PDF of the x -velocity u_1 at $(0.5, 0.4375)$ and u_2 at $(0.375, 0.4375)$: (a) direct simulation (b) probabilistic graphical model; the joint PDF of y -velocity v_1 at $(0.4375, 0.5)$ and v_2 at $(0.4375, 0.375)$: (c) direct simulation (d) probabilistic graphical model.

discussed in Section 4, the global posterior $p(\Theta|\mathcal{D})$ can be factorized into local posterior distributions for hyperparameters related to each coarse element. Then the local hyperparameters are learned from local training data \mathcal{D}_k which are local features and model responses on element E_k . This strategy relaxes the assumption of identical relationships between hidden variables and local features on different elements. However, more training data might be required to achieve sufficient accuracy in parameter learning. Here $N = 800, 1600, 2400$ training data are generated to train the probabilistic model. The source term f is set to be zero. Flow is induced from left to right side with Dirichlet boundary conditions $h = 1$ on $x = 0, h = 0$ on $x = 1$. No-flow Neumann boundary conditions are applied on the other two sides of the square domain.

As in Section 6.1, we first use the graphical model as a surrogate to predict model responses given randomly generated realization of the stochastic input. The probabilistic graphical model is trained with 2400 data points. The results are presented in Fig. 19. A 10-fold cross-validation is also performed with 1600 and 2400 samples. The cross-validation errors are shown in Fig. 20. Then we treat stochastic input as a random field with known probability distribution and perform belief propagation on the factor graph in Fig. 5(b) to predict the mean and variance of model responses. The number of kernels in nonparametric messages is set to be 4. The predictions are shown in Figs. 21–26. It is demonstrated in both examples that the accuracy of the probabilistic graphical model increases with the number of training data and the predictions converge to the reference solutions. In Fig. 27, using MC with 10^6 samples as the reference solution, the convergence plot shows that the graphical model prediction is less accurate than MC with 200 data points. This is expected because the parameter learning process is inaccurate with too small data sets. If the estimated hyperparameters are captured on local modes of posterior distributions and significantly deviate from the true values, the probabilistic graphical model may even give incorrect predictions. However, with the increase of the number of samples, the predictions of the probabilistic graphical model quickly converge.

The marginal PDFs of the model responses are also estimated with belief propagation in Figs. 28–30. Fig. 28 demonstrates that the accuracy of predictions can be improved by increasing the number of components in the messages in belief

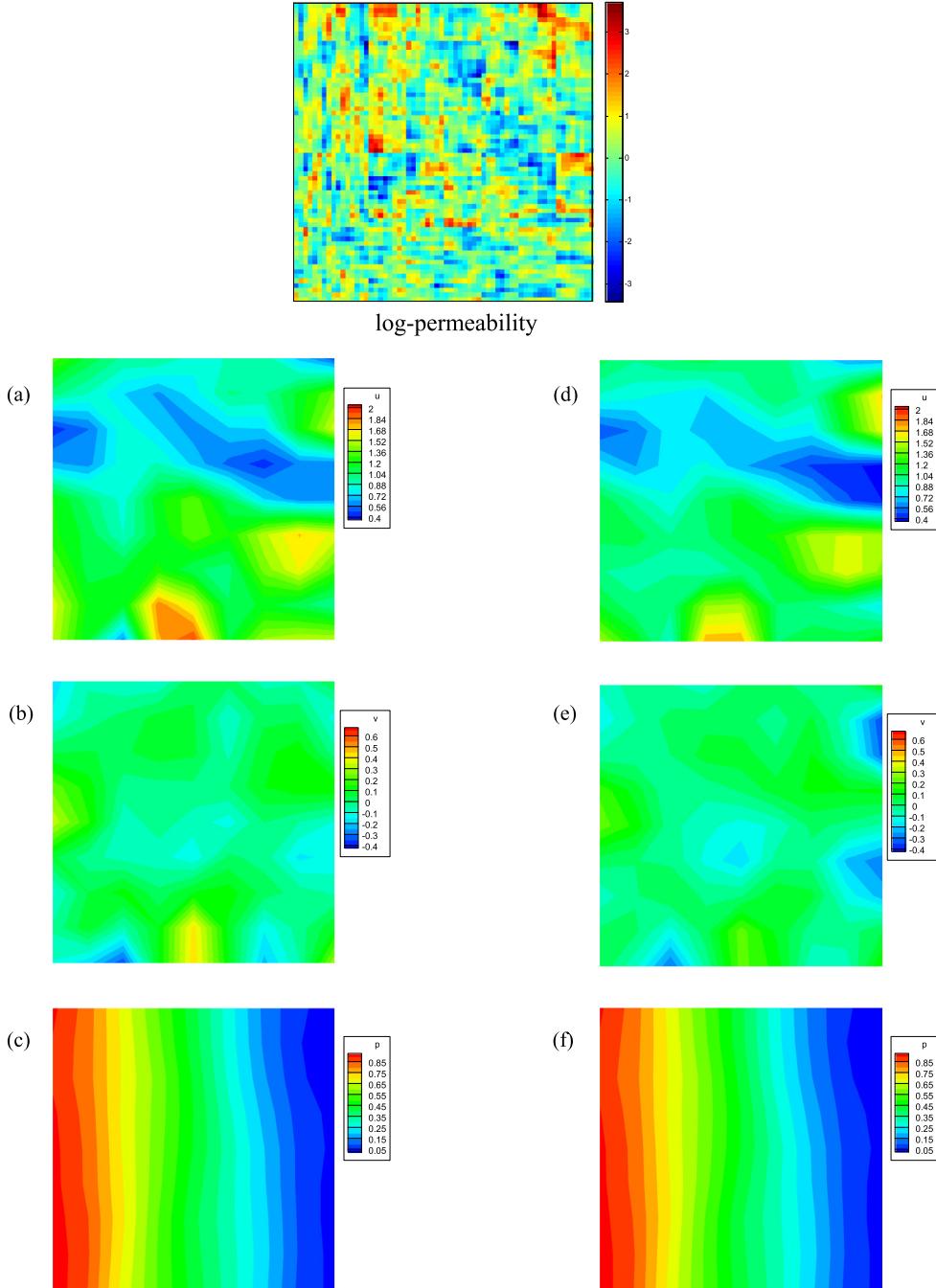


Fig. 32. Nonstationary random field: predicted values of model responses given a realization of stochastic input (a)–(c) x -velocity, y -velocity and pressure obtained from direct simulation, and (d)–(f) x -velocity, y -velocity and pressure predicted by the probabilistic graphical model (trained with 2400 data points).

propagation. However, it is achieved at the expense of increased computational cost. Although too few components in Gaussian mixtures is not enough to capture the shape of non-Gaussian PDFs, using excessive Gaussian components not only increases the time for message update but may also lead to overfitting. Fig. 29 shows the predicted PDFs of y -velocity at point $(0.4375, 0.5)$. We can see that the empirical PDF can be efficiently captured by two Gaussian kernels in this case. Further increasing the Gaussian components does not improve significantly the accuracy of prediction. Therefore, the choice of the number of components in nonparametric belief propagation should be made by taking a balance between the computational cost and accuracy of prediction. Fig. 30 shows the predicted PDFs of pressure on a coarse element centered at

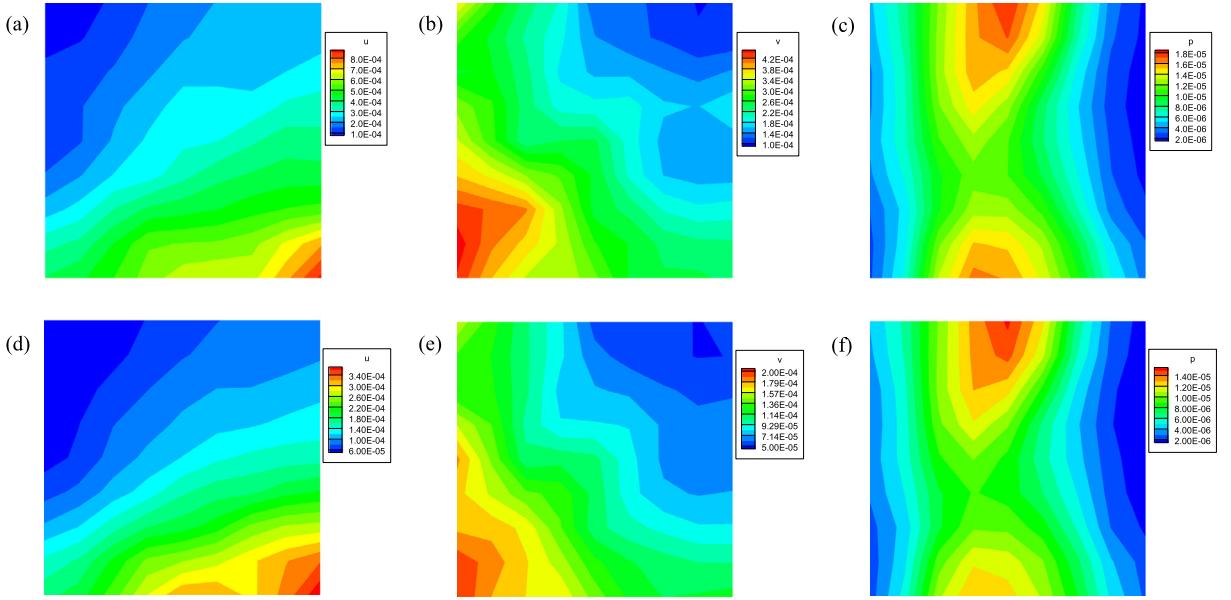


Fig. 33. Nonstationary random field: k -fold cross-validation error ($k = 10$) of x -velocity, y -velocity and pressure predicted by the probabilistic graphical model with (a)–(c) 1600 samples, and (d)–(f) 2400 samples.

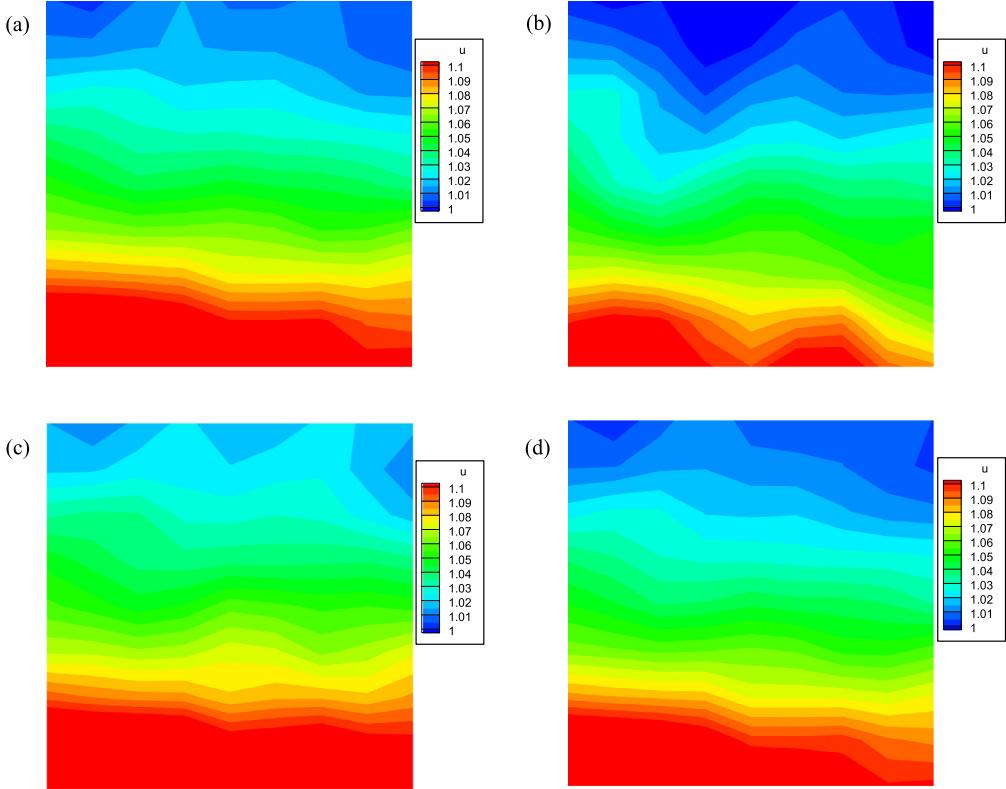


Fig. 34. Nonstationary random field: predicted mean of the x -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

(0.4375, 0.4375). In this case, two Gaussian components are sufficient to estimate the marginal PDF accurately as it is close to Gaussian. According to these results, the proper number of Gaussian components needed in nonparametric messages depends on the deviation of the target distribution from a Gaussian distribution. Finally, the joint distributions of x - and y -velocities at different locations are estimated with probabilistic graphical models trained by 2400 data points (Fig. 31).

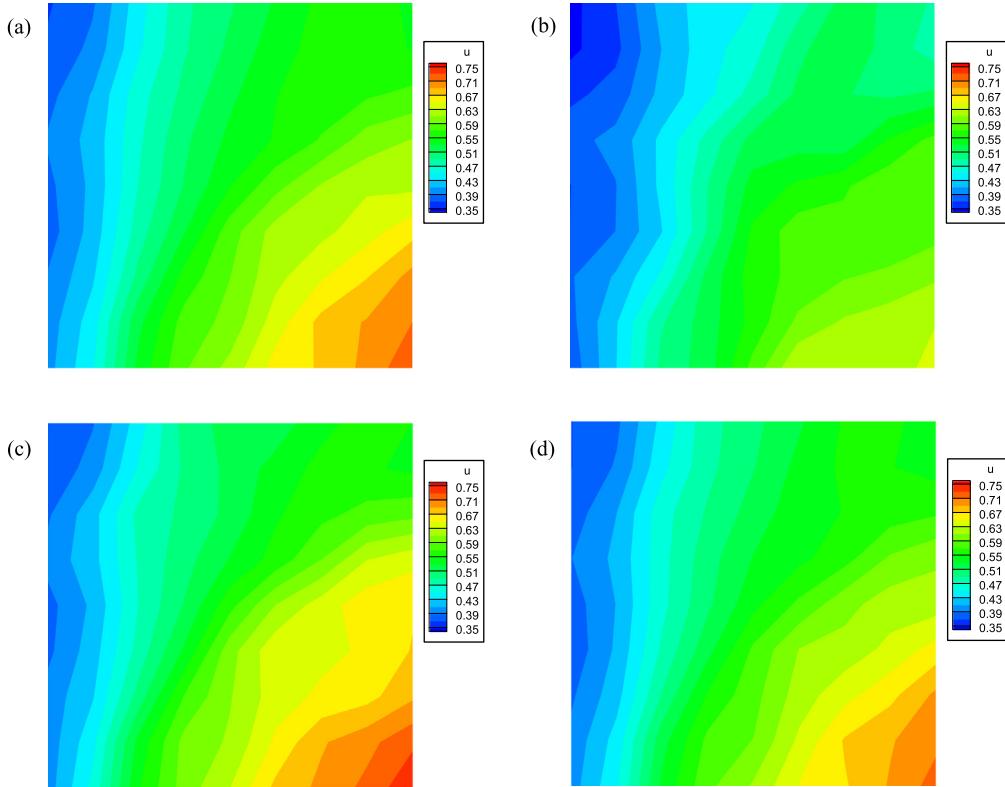


Fig. 35. Nonstationary random field: predicted variance of the x -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

6.3. Nonstationary random field

In the previous examples, it was assumed that the porous media are stationary such that the covariance between any two points in the domain depends on their distance rather than their actual locations. However, hydraulic properties may exhibit spatial variations at various scales. Therefore, it is important to extend the probabilistic graphical model to nonstationary random fields. In this example, we use a nonstationary random field as stochastic input. The log-permeability on the k th coarse element is a Gaussian random field with mean zero and an exponential covariance function

$$\text{cov}(\mathbf{x}, \mathbf{x}^*) = \sigma^2 \exp\left(-\frac{|\mathbf{x}_1 - \mathbf{x}_1^*|}{L_{k,1}} - \frac{|\mathbf{x}_2 - \mathbf{x}_2^*|}{L_{k,2}}\right). \quad (42)$$

The correlation lengths $L_{k,1}$ and $L_{k,2}$ vary on the coarse scale. Since the coarse grid has $N_x = 8$ rows and $N_y = 8$ columns of elements, we define the coordinate of the k th element as (i_k, j_k) where i_k is the index in row and j_k is the index in column. Then the correlation length is set to be $L_{k,1} = 0.1 + \frac{0.4}{N_y-1} j_k$ and $L_{k,2} = 0.1 + \frac{0.4}{N_x-1} i_k$. The source term and boundary conditions are the same as those in Section 6.2.

The challenge of a nonstationary random field is that the influence of local properties on local responses could vary on coarse elements as the correlation between hydraulic properties depends on the location. In this case, it is difficult to estimate the probabilistic model globally due to the large number of hyperparameters. The probabilistic graphical model proposed in Section 3.2 efficiently decomposes the global problem into local lower dimensional problems. In this way, we can estimate the hyperparameters locally with local posterior distributions defined in Section 4.

In this example, $N = 800, 1600, 2400$ training data are generated to train the probabilistic model. The belief propagation runs in the same way as in previous examples. Figs. 32 and 33 verify the correctness of the probabilistic model in its ability as a surrogate model. The estimated mean and variance of model responses are shown in Figs. 34–39. The convergence plot is presented in Fig. 40. Compared with the convergence plot in the example of Section 6.2, it is seen that the convergence rate is not significantly affected by the stationarity/nonstationarity. This is because the hyperparameters are learned locally, which implies that the convergence depends on the number of data sets and the number of hyperparameters in each coarse element. Figs. 41–43 show the predicted marginal PDFs of model responses. Fig. 44 shows the joint distributions of model responses at different locations estimated with a probabilistic graphical model trained by 2400 data points. With the local

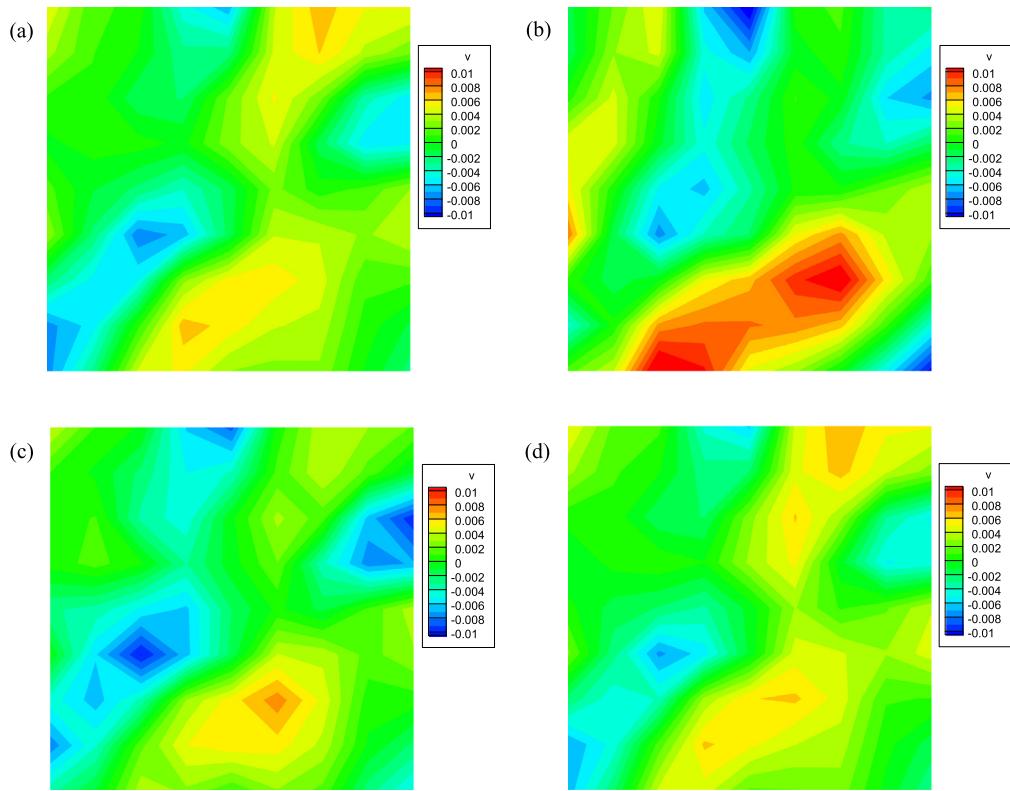


Fig. 36. Nonstationary random field: predicted mean of the y -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

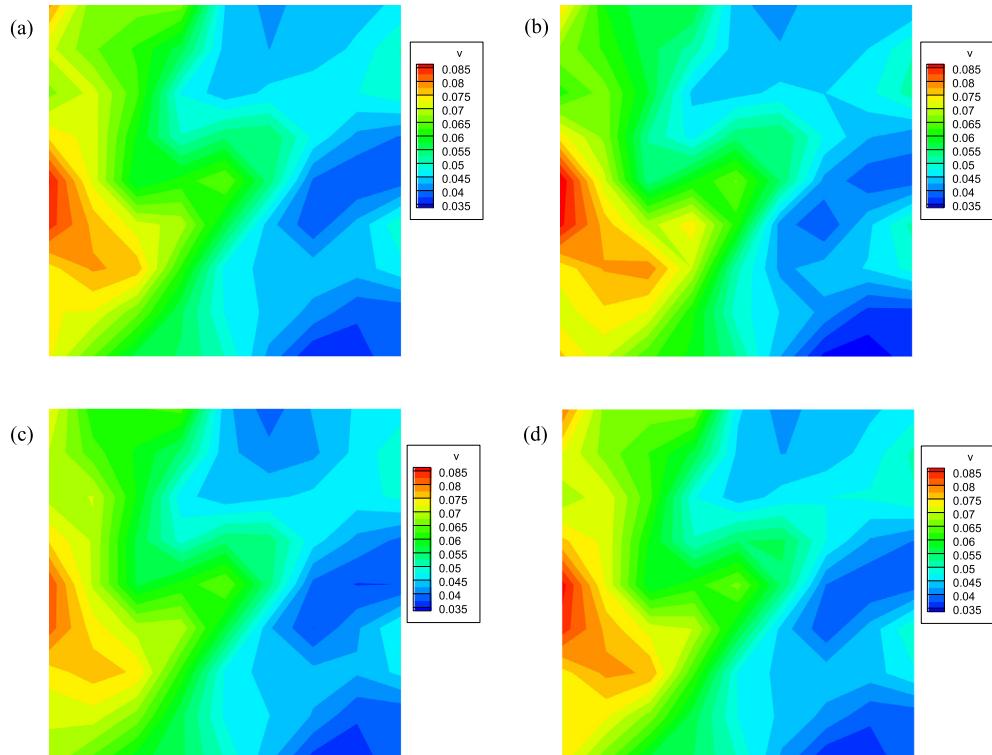


Fig. 37. Nonstationary random field: predicted variance of the y -velocity from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

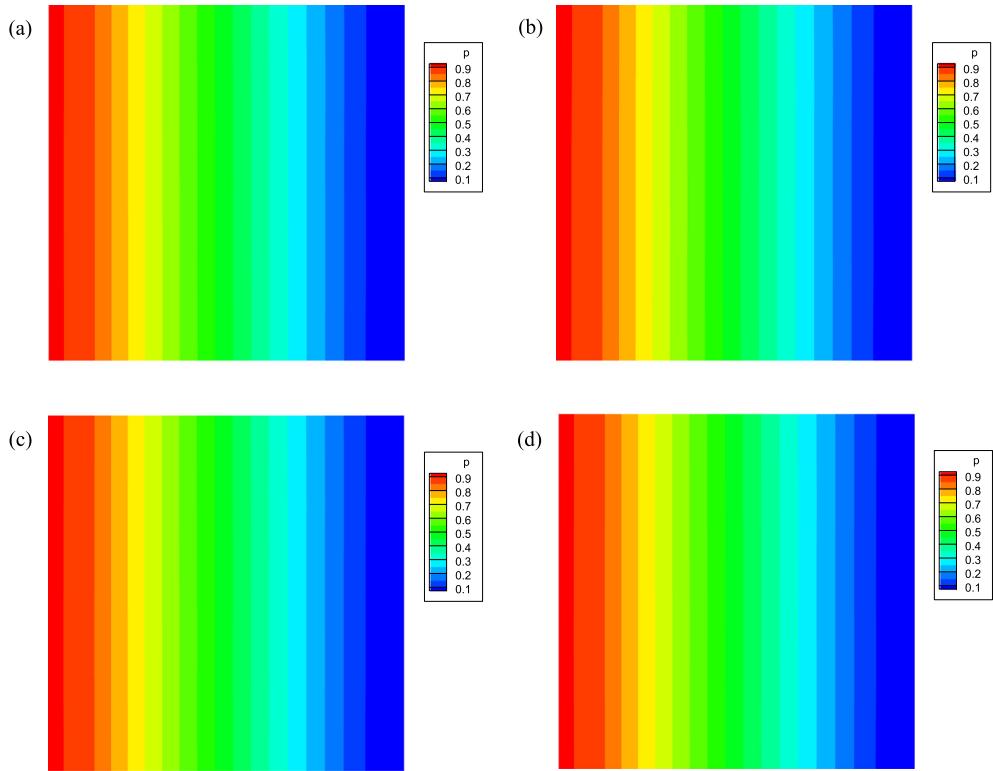


Fig. 38. Nonstationary random field: predicted mean of pressure from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

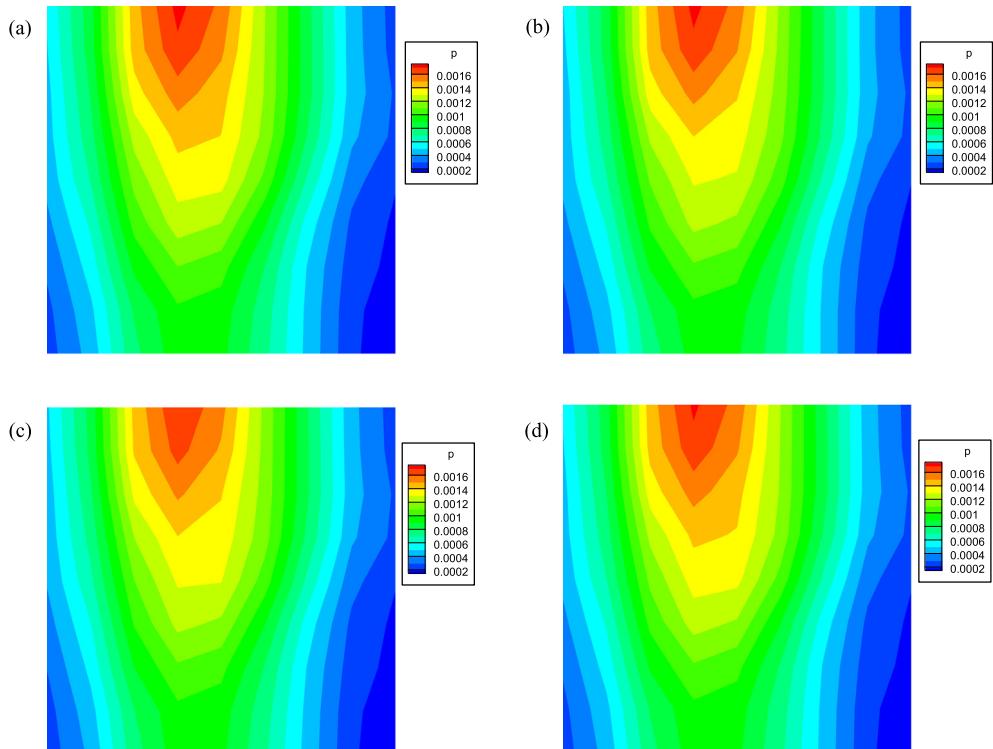


Fig. 39. Nonstationary random field: predicted variance of pressure from (a) MC simulation, and from probabilistic graphical models trained by (b) 800, (c) 1600 and (d) 2400 data.

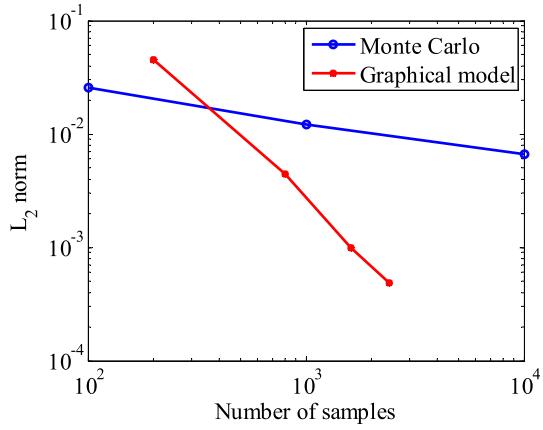


Fig. 40. Nonstationary random field: the L_2 norm of the error in the variance of flux as a function of the observed samples for MC simulation and graphical model prediction.

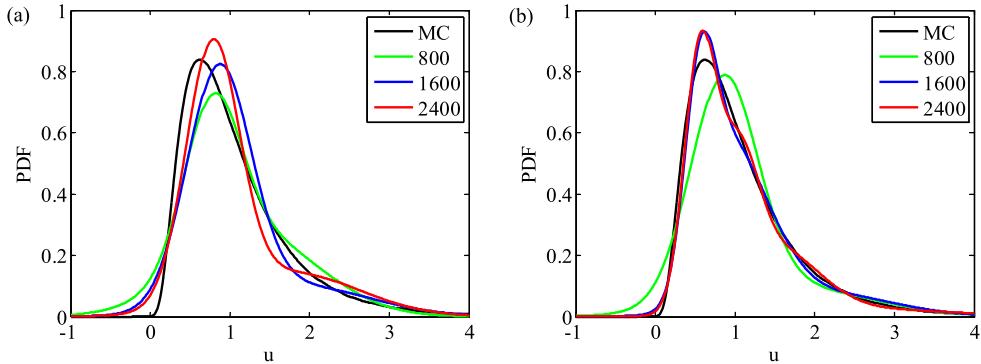


Fig. 41. Nonstationary random field: predicted marginal PDF of the x -velocity at point $(0.5, 0.4375)$: Using (a) 2 and (b) 4 Gaussian components in nonparametric messages.

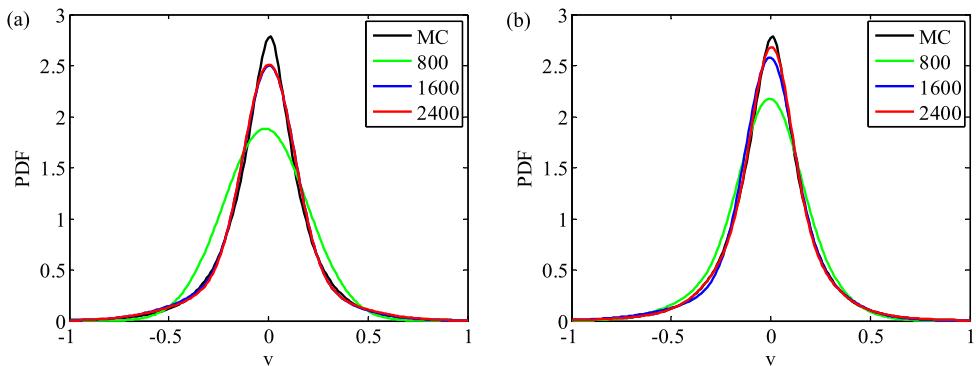


Fig. 42. Nonstationary random field: predicted marginal PDF of the y -velocity at point $(0.4375, 0.5)$: Using (a) 2 and (b) 4 Gaussian components in nonparametric messages.

hyperparameter learning strategy, this example requires no more training data than the last example in which the stochastic input is a stationary random field. On the other hand, it is suggested to use global parameter learning for stationary random field as hyperparameters on different coarse elements are identical. According to the first example in Section 6.1, a relatively small number of training data is sufficient to give accurate probabilistic models.

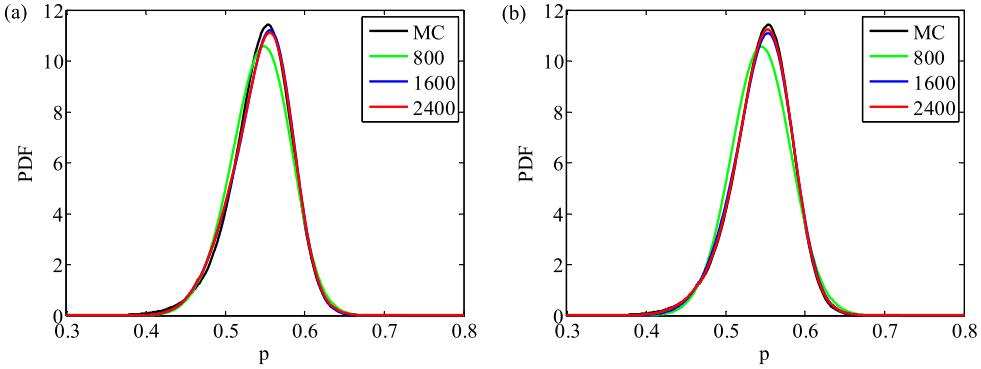


Fig. 43. Nonstationary random field: predicted marginal PDF of pressure at the coarse element centered at point (0.4375, 0.4375): Using (a) 2 and (b) 4 Gaussian components in nonparametric messages.

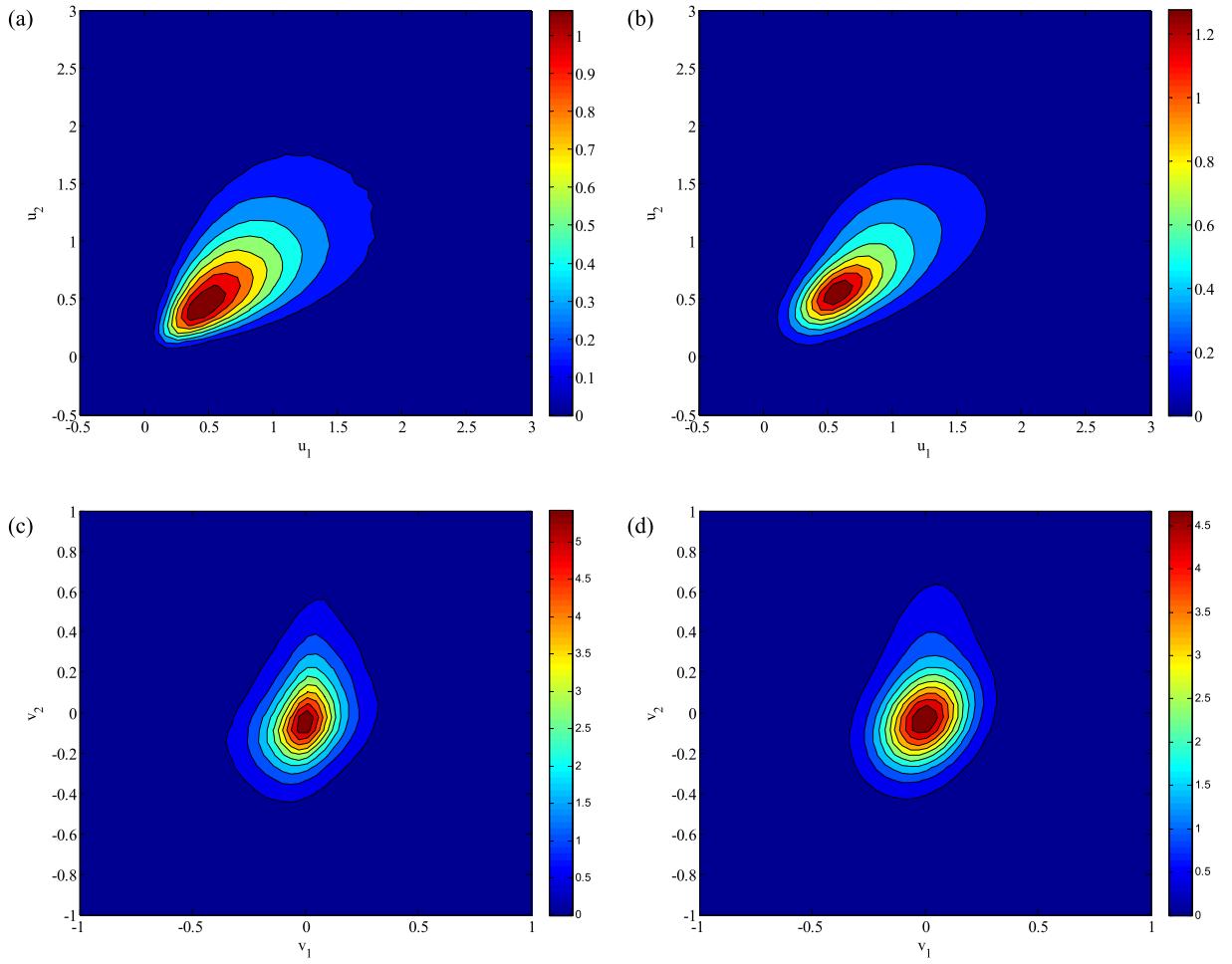


Fig. 44. Nonstationary random field: the joint PDF of the x -velocity u_1 at (0.5, 0.4375) and u_2 at (0.375, 0.4375); (a) direct simulation (b) probabilistic graphical model; the joint PDF of the y -velocity v_1 at (0.4375, 0.5) and v_2 at (0.4375, 0.375); (c) direct simulation (d) probabilistic graphical model.

7. Conclusions

In this paper, a probabilistic graphical model is constructed to approximate the solutions to multiscale SPDEs. The basic idea is to treat the stochastic input as well as model responses as random variables and to represent their relationships explicitly using a factor graph model. Thus the high-dimensional joint distribution can be factorized by the potential

functions that describe the interactions among neighboring variables in the graph. In order to relieve the curse of dimensionality, a set of hidden variables defined on a coarse-scale are employed to bridge fine-scale features and coarse-scale responses. The graphical model not only facilitates probabilistic modeling of model responses but also enables us to solve the inference problem efficiently with the help of belief propagation algorithms. These algorithms marginalize unobserved random variables in a graphical model by propagating special functions, messages, between variables (including variable nodes and factor nodes in a factor graph).

One of the most important issues in the belief propagation algorithm with nonparametric messages is the specification of number of Gaussian components in the nonparametric messages. Numerical results show that insufficient number of components will lead to inaccurate predictions, while excessive components will increase computational cost or even lead to overfitting. The proper number of components mainly depends on the deviation of target marginal distributions from Gaussian distributions.

It is also straightforward to extend the current graphical model to incorporate higher-order interactions between random variables. Note that the interactions between variables are explicitly denoted by the factor nodes in the factor graph in Fig. 3(b). Each factor node is a potential function and connects all its member variable nodes. When a high-order potential function is considered, one just adds a corresponding factor node in the factor graph and connects it with all its member variables. The belief propagation algorithm introduced in Section 5 strictly follows the update rules in Eqs. (31) and (32) which in theory can deal with any-order potential functions. Thus we can use the same algorithm to make inference on a factor graph with high-order potentials. However, high-order interactions can lead to more complicated graph structure and high-dimensional message propagation. As a result, the computational cost would be significantly increased.

Acknowledgments

This research was supported by an OSD/AFOSR MURI09 award on uncertainty quantification, the US Department of Energy, Office of Science, Advanced Scientific Computing Research and the Computational Mathematics program of the National Science Foundation (NSF) (award DMS-1214282). This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the US Department of Energy under Contract No. DE-AC02-05CH11231. Additional computing resources were provided by the NSF through TeraGrid resources provided by NCSA under Grant No. TG-DMS090007.

References

- [1] A. Nouy, Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations, *Archives of Computational Methods in Engineering* 16 (3) (2009) 251–285.
- [2] J. Hesthaven, E. Ronquist, *Spectral and High Order Methods for Partial Differential Equations: Selected Papers from the ICOSAHOM '09 Conference, June 22–26, Trondheim, Norway*, Lecture Notes in Computational Science and Engineering, Springer, 2010.
- [3] S.A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, in: *Dokl. Akad. Nauk SSSR*, vol. 4, 1963, p. 123.
- [4] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, *SIAM Journal on Scientific Computing* 27 (3) (2005) 1118–1139.
- [5] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, *Communications in Computational Physics* 2 (2) (2007) 293–309.
- [6] F. Nobile, R. Tempone, C. Webster, A sparse grid collocation method for elliptic partial differential equations with random input data, *SIAM Journal of Numerical Analysis* 45 (2008) 2309–2345.
- [7] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, *Journal of Computational Physics* 228 (8) (2009) 3084–3113.
- [8] X. Ma, N. Zabaras, An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations, *Journal of Computational Physics* 229 (10) (2010) 3884–3915.
- [9] A. Doostan, G. Iaccarino, A least-squares approximation of partial differential equations with high-dimensional random inputs, *Journal of Computational Physics* 228 (12) (2009) 4332–4345.
- [10] X. Ma, N. Zabaras, A stochastic mixed finite element heterogeneous multiscale method for flow in porous media, *Journal of Computational Physics* 230 (12) (2011) 4696–4722.
- [11] G. Li, C. Rosenthal, H. Rabitz, High dimensional model representations, *The Journal of Physical Chemistry A* 105 (33) (2001) 7765–7777.
- [12] X. Yang, M. Choi, G. Lin, G.E. Karniadakis, Adaptive anova decomposition of stochastic incompressible and compressible flows, *Journal of Computational Physics* 231 (4) (2012) 1587–1614.
- [13] T.Y. Hou, X. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, *Journal of Computational Physics* 134 (1997) 169–189.
- [14] T.Y. Hou, X.-H. Wu, Z. Cai, Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients, *Mathematics of Computation* 68 (1999) 913–943.
- [15] Y. Efendiev, T.Y. Hou, *Multiscale Finite Element Methods: Theory and Applications*, Surveys and Tutorials in the Applied Mathematical Sciences, Springer, Dordrecht, 2009.
- [16] T.J.R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, *Computer Methods in Applied Mechanics and Engineering* 127 (1995) 387–401.
- [17] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method – a paradigm for computational mechanics, *Computer Methods in Applied Mechanics and Engineering* 166 (1998) 3–24.
- [18] E. Weinan, B. Engquist, The heterogeneous multi-scale methods, *Communications in Mathematical Sciences* 1 (2002) 87–132.
- [19] E. Weinan, B. Engquist, X. Li, W. Ren, E. Vanden-Eijnden, Heterogeneous multiscale methods: a review, *Communications in Computational Physics* 2 (2007) 367–450.
- [20] W.L. Wan, T.F. Chan, B. Smith, An energy-minimizing interpolation for robust multigrid methods, *SIAM Journal on Scientific Computing* 21 (4) (1999) 1632–1649.
- [21] Y. Efendiev, J. Galvis, P. Vassilevski, Spectral element agglomerate algebraic multigrid methods for elliptic problems with high-contrast coefficients, in: Y. Huang, R. Kornhuber, O. Widlund, J. Xu (Eds.), *Domain Decomposition Methods in Science and Engineering XIX, Lecture Notes in Computational Science and Engineering*, vol. 78, Springer, Berlin, Heidelberg, 2011, pp. 407–414.

- [22] M.J. Wainwright, M.I. Jordan, Graphical models, exponential families, and variational inference, *Foundation and Trends in Machine Learning* 1 (1–2) (2008) 1–305.
- [23] C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, first ed., Springer, 2007.
- [24] J.E. Aarnes, V. Kippe, K.-A. Lie, A.B. Rustad, Modelling of Multiscale Structures in Flow Simulations for Petroleum Reservoirs, in: G. Hasle, K.-A. Lie, E. Quak (Eds.), *Geometric Modelling, Numerical Simulation, and Optimization*, Springer, Berlin, Heidelberg, 2007, pp. 307–360 (Ch. 10).
- [25] H. Rue, L. Held, *Gaussian Markov Random Fields: Theory and Applications*, Monographs on Statistics and Applied Probability, vol. 104, Chapman & Hall, London, 2005.
- [26] F.R. Kschischang, B.J. Frey, H.-A. Loeliger, Factor graphs and the sum-product algorithm, *IEEE Transaction on Information Theory* 47 (2001) 498–519.
- [27] J. Besag, Spatial Interaction and the Statistical Analysis of Lattice Systems, *Journal of the Royal Statistical Society. Series B (Methodological)* 36 (2) (1974) 192–236.
- [28] S. Lauritzen, *Graphical Models*, Oxford Science Publications, Clarendon Press, 1996.
- [29] I. Nabney, *NETLAB: Algorithms for Pattern Recognition, Advances in Pattern Recognition Series*, Springer, 2004.
- [30] S. Chakraborty, Some applications of Dirac's delta function in statistics for more than one random variable, *Applications and Applied Mathematics* 3 (1) (2008) 42–54.
- [31] P.D. Moral, A. Doucet, A. Jasra, Sequential Monte Carlo samplers, *Journal of the Royal Statistical Society: Series B* 68 (3) (2006) 411–436.
- [32] J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, New York, 2008.
- [33] A. Billy, F.Y. Bois, E. Parent, C.P. Robert, Bayesian-optimal design via interacting particle systems, *Journal of the American Statistical Association* 101 (474) (2006) 773–785.
- [34] A. Jasra, A. Doucet, D.A. Stephens, C.C. Holmes, Interacting sequential Monte Carlo samplers for trans-dimensional simulation, *Computational Statistics and Data Analysis* 52 (4) (2008) 1765–1791.
- [35] J. Wan, N. Zabaras, A Bayesian approach to multiscale inverse problems using the sequential Monte Carlo method, *Inverse Problems* 27 (10) (2011) 105004.
- [36] P.S. Koutsourelakis, A multi-resolution, non-parametric, Bayesian framework for identification of spatially-varying model parameters, *Journal of Computational Physics* 228 (17) (2009) 6184–6211.
- [37] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing* 10 (3) (2000) 197–208.
- [38] J.S. Liu, R. Chen, Blind deconvolution via sequential imputations, *Journal of the American Statistical Association* 90 (430) (1995) 567–576.
- [39] J.S. Liu, R. Chen, Sequential Monte Carlo methods for dynamic systems, *Journal of the American Statistical Association* 93 (443) (1998) 1032–1044.
- [40] J.S. Yedidia, W.T. Freeman, Y. Weiss, Understanding belief propagation and its generalizations, Tech. Rep. TR-2001-22, Mitsubishi Electric Research Laboratories, Jan. 2002.
- [41] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [42] J.S. Yedidia, W.T. Freeman, Y. Weiss, Generalized belief propagation, in: IN NIPS 13, vol. 13, 2000, pp. 689–695.
- [43] K.P. Murphy, Y. Weiss, M.I. Jordan, Loopy belief propagation for approximate inference: an empirical study, in: *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, UAI'99*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, pp. 467–475.
- [44] E.B. Sudderth, A.T. Ihler, M. Isard, W.T. Freeman, A.S. Willsky, Nonparametric belief propagation, *Communications of the ACM* 53 (10) (2010) 95–103.
- [45] G. McLachlan, D. Peel, *Finite Mixture Models*, first ed., Wiley Series in Probability and Statistics, Wiley-Interscience, 2000.
- [46] P. Raviart, J. Thomas, A mixed finite element method for 2nd order elliptic problems, in: E. Magenes Galligani (Ed.), *Mathematical Aspects of Finite Element Methods*, Lecture Notes in Mathematics, vol. 606, Springer-Verlag, 1977, pp. 292–315.
- [47] D.N. Arnold, Mixed finite element methods for elliptic problems, *Computer Methods in Applied Mechanics and Engineering* 82 (1–3) (1990) 281–300.
- [48] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI'95*, vol. 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143.